Industry Foundation Classes - Release 2.0

Specifications Volume 1

AEC/FM Processes Supported By IFC

Final Release --15-March-99

International Alliance for Interoperability
Enabling Interoperability in the AEC/FM Industry

Industry Foundation Classes - Release 2.0

Specifications Volume 1

AEC/FM Processes supported by IFC

Enabling Interoperability in the AEC/FM Industry

Mailing address: 2960 Chain Bridge Road - Suite 143
Oakton, Virginia 22124

Email address: IAI@Interoperability.com

Web Address: www.Interoperability.com

Document Editor

Editor	Richard See
Development committee	Specification Task Force

Document Control

Project reference	IFC Release 2.0
Document reference	AEC/FM Processes supported by IFC
Document version	IFC Release 2.0 – Final
Release date	March 15, 1999
Status	For Comments
Distribution	IAI Member Companies
Distribution format	PDF file

Revisions

Rev.	Person	Date	Description
Alpha	Richard See	10-Aug-98	Alpha release
Beta	Richard See	10-Jan-99	Beta release
Final	Richard See	15-Mar-99	Final release

Table of Contents

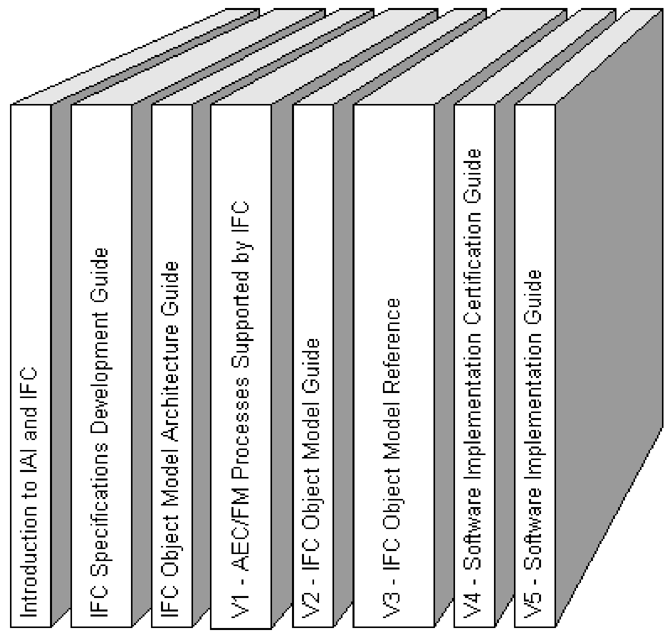
1. Introduction, Scope and Assumptions

1.1. Purpose of these documents

The purpose of this document suite is to provide a detailed specification of the Industry Foundation Classes (IFC) as defined by the Industry Alliance for Interoperability (IAI). The intended audience is the IAI membership, industry domain experts, and software developers interested in implementing IFC.

1.2. IFC Release Document Suite

IFC will be documented for two readers. The AEC professional and the software profession serving the AEC industry. Documents in this release include:



An Introduction to IAI and IFC

The "*An Introduction to IAI and IFC,"* as the name implies, provides AEC/FM industry professionals with an introduction to the organization, including its mission and organization. It also introduces the shared project model concept, end user benefits in using IFC compliant applications and summarizes the AEC Industry processes that are supported by this release of IFC. Finally, it provides a preview of what will be added in future releases.

IFC Specification Development Guide

The *"IFC Specification Development Guide"* defines the process used by the IAI in developing IFC. It also provides various references supporting parts of this process such as development of process diagrams, development of detailed requirement definitions and reading/creating EXPRESS (data model) definitions and EXPRESS-G diagrams.

IFC Object Model Architecture Guide

The *"IFC Object Model Architecture* G*uide"* defines the architecture used in the design of the IFC object model. This architecture is modular and layered which allows independent development and evolution of sub-schemata. This document is written for software developers who will develop applications supporting IFC.

Volume 1: AEC/FM Processes Supported by IFC

THIS DOCUMENT -- The "*AEC/FM Processes Supported by IFC"* volume documents the AEC/FM industry processes that the IFC Project Model in this release is designed to support. Therefore, this document effectively defines the scope of AEC project information included in this Release. Volumes 2 and 3 structure this information for use in applications. Note that this IFC release is limited to the information content of the foundation classes defined. Behavior for these objects, and thus the implementation of software that will support these AEC industry processes, will be defined by the implementing software vendors.

Volume 2: IFC Object Model Guide

The "*IFC Object Model Guide"* defines model design and use concepts for IFC object model. These key concepts include: an overview of model architecture, capturing design intent, sharing semantic relationships, model extension by application developers. It also describes some implementation strategies such as file based model exchange, Client-Server architectures and runtime interoperability supported through standard software interfaces of the IFC model. This includes provides an overview and example of the physical file format for file based model exchange.

Volume 3: IFC Object Model Reference

The "*IFC Object Model Reference"* provides detailed definitions for each of the classes and data types defined in the IFC object model. This includes all of the information required by the AEC processes defined in volume 1, structured in an information model detailing object class data, relationships, standard interfaces, type definitions and geometry schema use for shape representation. Additionally, it provides a data model view defined in EXPRESS and a standard interfaces view defined in IDL. Each of these code sets will be used by application developers as input into Computer Aided Software Engineering (CASE) tools to semi-automate development of applications supporting IFC. Finally, a on-line version of this information is provided using an HTML document set that is cross linked for easy access to information related to or supporting a particular class or data type.

Volume 4: IFC Software Implementation Certification Guide

The "*IFC Software implementation Certification Guide"* provides detailed information about conformance certifications issues and the methodology that will be used by the IAI to certify applications for multiple levels of IFC conformance. This includes an overview of the concepts for conformance assessment and certification, definition of various "Exchange Set" subsets of the IFC model for which certification can be assessed and an overview of the testing suites that will be used for certification testing.

Volume 5: IFC Software Implementation Guide

The "*IFC Software implementation Guide"* provides detailed information addressing the issues of implementing the IFC object model in software products. In this release, it's content is limited to the topics of implementing property sets (previously called "Pset Guide") and the differences from the previous release (previously called "Migration Guide"). Over the next couple of IFC releases, many more topics will be addressed.

1.3. Scope

1.3.1. Scope for IFC Release 2.0

Enabling interoperability between applications by different software vendors is the ultimate goal of the IAI. This is a very ambitious goal and will be achieved through a series of incremental steps.

In general, the IAI is focused on providing three things in IFC:

1. Standard definitions for the attributes associated with entities comprising an AEC/FM project model (objects)
2. Structure and relationships between these entities from the point of view of various AEC/FM professionals
3. Standard formats/protocols for two methods of sharing this information:
    - *exchange via a standard file format*
    - *exchange via standard software interfaces*

It is important to note that the software interface specifications in this release will not include any application-specific behavior. Instead, these interfaces will be limited to get and set methods for the attribute and relationship information defined in the data model.

Release 1.5 of IFC provided the infrastructure that supports this release, plus reasonable models for architecture, some HVAC, estimating, scheduling and Facilities Management. This release will build on these foundations and extend the model in several areas.

The scope for this release of the IFC Specifications is limited to:

1. Six AEC/FM domains - Architecture, HVAC engineering, codes and standards, cost estimating, facilities management and simulation
2. Only a specific subset of the processes in these domains (defined in Volume 1 of these specifications).

These domains and processes are:

Architectural Design

- *Building 'shell' design*
- *Building 'core' design*
  - *Stair design*
  - *Public toilet design*
- *Roof design*
- *Fire Compartmentation*

HVAC Engineering

- *HVAC Duct System Design*
- *HVAC Piping System Design*
- *Pathway Design and Coordination*
- *Building Heating and Cooling Load Calculation*

Codes and Standards

- *Commercial and Residential Energy Code Compliance Checking*
- *Handicapped access code checking*
- *Escape from Fire code checking*

Cost Estimating

- *Cost Estimating*
  - *Identify Objects*
  - *Identify Tasks Needed to Install Objects*
  - *Identify Resources Needed to Perform Tasks*
  - *Quantify*
  - *Costing and Cost Summarization*

Facilities Management

- *Property Management*
  - *Enabling the use of IFC objects in property management*
  - *Grouping IFC objects*
  - *Linking the maintenance objects to the IFC objects*
- *Occupancy Planning*
- *Design of Workstations*
- *Floor Layout of Workstations for an Open Office*

Simulation

- *Photo Accurate Visualization*

All AEC domains

- *Document references (from model to document only)*

1.3.2. Scope of this document

This document includes the following information:

1. **Introduction, Scope and Assumptions**

   This section provides the reader with an introduction to the set of seven documents comprising this release of the IFC Specifications. This section outlines the information included in this document versus related documents. It will also define the scope for this release and assumptions about knowledge of the reader.

2. **AEC/FM Process Framework**

   This section provides an overview of the AEC industry processes that are performed through out the design, engineer, build, and management of a built facility. The diagrams are meant to be a framework for the reader of these documents to provide an orientation for indicating where a process fits into the building lifecycle. Processes defined and supported in previous releases are indicated as gray shaded process boxes. Processes defined in this release are indicated with by black shaded process boxes.

3. **Domain Team Project Summaries**

   This section provides the reader with an overview of the AEC/FM domain projects that developed the requirements for this release. An description of the project team, and overview of the industry processes for which requirements are defined and an overview of the resources required for the project are provided.

4. **AEC Process Definitions and Usage Scenarios**

   This section includes the process definitions and usage scenarios which are the basis for the information requirements specified in the next section - and ultimately, for the extensions to the IFC model in this release. The specified processes were prioritized and selected as processes that would see significant improvements (efficiency, cost avoidance, etc.) if supported by IFC. There are a few criteria used to do this. First, IFC support for the process must provide an increase in productivity and must be concise enough to be completed in a single IFC release cycle. Second, the process should deliver a benefit to other domains in the building life cycle. Third, there must be a minimum of two software companies that have committed that they will implement support for the process and associated IFC objects in a shipping software product.

   Such processes obviously vary between companies and certainly between regions. The definitions specified represent the IAI domain groups' consensus on a generalized definition that sufficiently represents the diversity across companies and regions. It is anticipated that future releases of IFC will reflect some regional differences.

   Each process in this section contains three parts. The first provides an overview process description, written to AEC professionals, to indicate where the process fits into their overall processes. The second part is a process diagram which illustrates each task in the process and its informational input/output sources. The third part provides text book style task definitions and a running series of usage scenarios using real project graphics and data. These are organized according to the tasks in the process diagram.

5. **Information Requirement Analyses**

   This section provides a detailed analysis for all of the input information required and output information supplied by each of the process tasks defined in the previous section.

6. **Object Type Definition Tables**

   This section organizes information requirements by 'object type' and provides detailed information about data types, limits and defaults for all information.

1.4. Assumptions and Abbreviations

This document assumes the reader is reasonably familiar with the following:

- AEC/FM market and project terminology
- Software industry terminology
- Concepts and terminology associated with object oriented software

The following abbreviations are used throughout the IFC Specifications:

- AEC/FM Architectural, Engineering, Construction and Facilities Management
- IAI Industry Alliance for Interoperability
- AP Application Protocol
- Arch Architecture
- CM Construction Management
- CORBA Common Object Request Broker Architecture
- COM Microsoft's Component Object Model
- DCE Distributed Computing Environment
- DCOM Microsoft's Distributed Component Object Model
- DSOM IBM's Distributed System Object Model
- FM Facilities Management
- FTP File Transfer Protocol
- GUID Globally Unique Identifier
- HVAC Heating, Ventilating and Air Conditioning
- HTTP Hypertext Transport Protocol
- IAI International Alliance for Interoperability
- IDL Interface Definition Language
- IFC Industry Foundation Classes
- ISO International Standards Organization
- FM Facilities Management
- MIDL Microsoft's Interface Definition Language
- ODL Microsoft's Object Description Language
- OMG Object Management Group
- ORB Object Request Broker
- OSF Open Software Foundation
- RPC Remote Procedure Call
- SOM IBM's System Object Model
- STEP Standard for the Exchange of Product Model Data
- TCP/IP Transmission Control Protocol/Internet Protocol
- TQM Total Quality Management
- URL Universal Resource Location

1.5. International Alliance for Interoperability (IAI)

The IAI is a 'not for profit' industry alliance of companies. Its membership is comprised of visionary companies representing all sectors of the AEC industry worldwide.

The IAI was first formed in September of 1995, by 12 industry leading companies who, during the previous year had worked together to develop proof of concept prototypes demonstrating the viability of interoperability between AEC software applications. This demonstration was shown publicly at the AEC Systems '95 conference in Atlanta, Georgia. This is the third release of IFC since that time. There are currently 50 organizations implementing software to support IFC, a number that is growing quite rapidly now.

As of this printing, the IAI includes 9 international chapters with hundreds of member companies in the following regions:

- *Australasian countries*
- *French speaking region of Europe*
- *German speaking region of Europe*
- *Japan*
- *Korea*
- *Nordic countries of Europe*
- *North America*
- *Singapore*
- *United Kingdom*

The IAI stated Vision, Mission and Values can be summarized as:

VISION

Enabling Interoperability in the A/E/C/FM Industry

MISSION

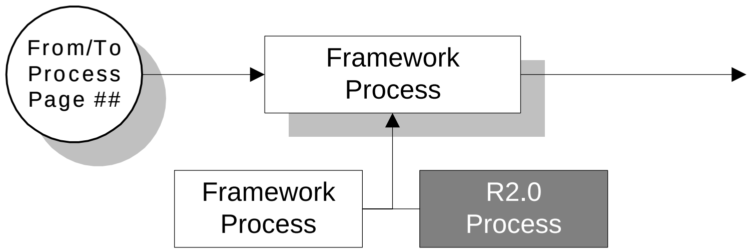
To define, promote and publish specifications for the Industry Foundation Classes (IFC) as a basis for information sharing through the project life cycle, globally, across disciplines and technical applications.

VALUES

- Not for profit industry organization
- Action oriented (Alliance v. Association)
- Consensus based decision making
- Incremental delivery (rather than prolonged study)
- Global solution
- Industry to define IFC
- IFC to be "open" (for implementation/use by all software vendors)
- Design for IFC to be extensible
- IFC will evolve over time
- Membership open to any company working in construction industry

2. AEC/FM Industry Process Framework

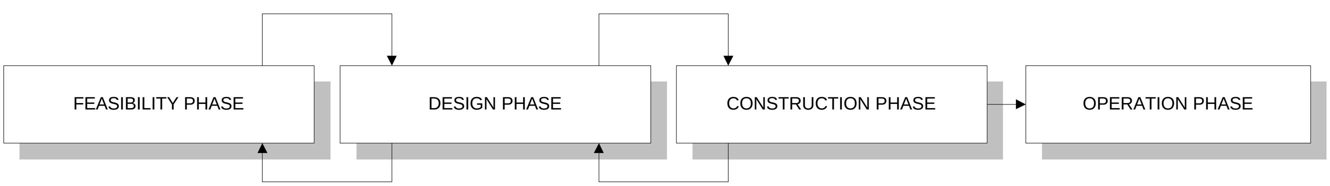
This section includes diagrams expressing a high level framework for the AEC/FM industry processes. Three levels of diagrams are provided. The first diagram represents the four phases of a project as an index into the decision to study, design, construct, and operate a facility through its entire lifecycle. The second set of diagrams presents each phase as a series of processes that are accomplished by participating disciplines. The third set of diagrams provide a process breakdown (sub-processes). Processes supported by this release of IFC are highlighted. Those that are not highlighted will be considered in future IFC releases.



KEY:

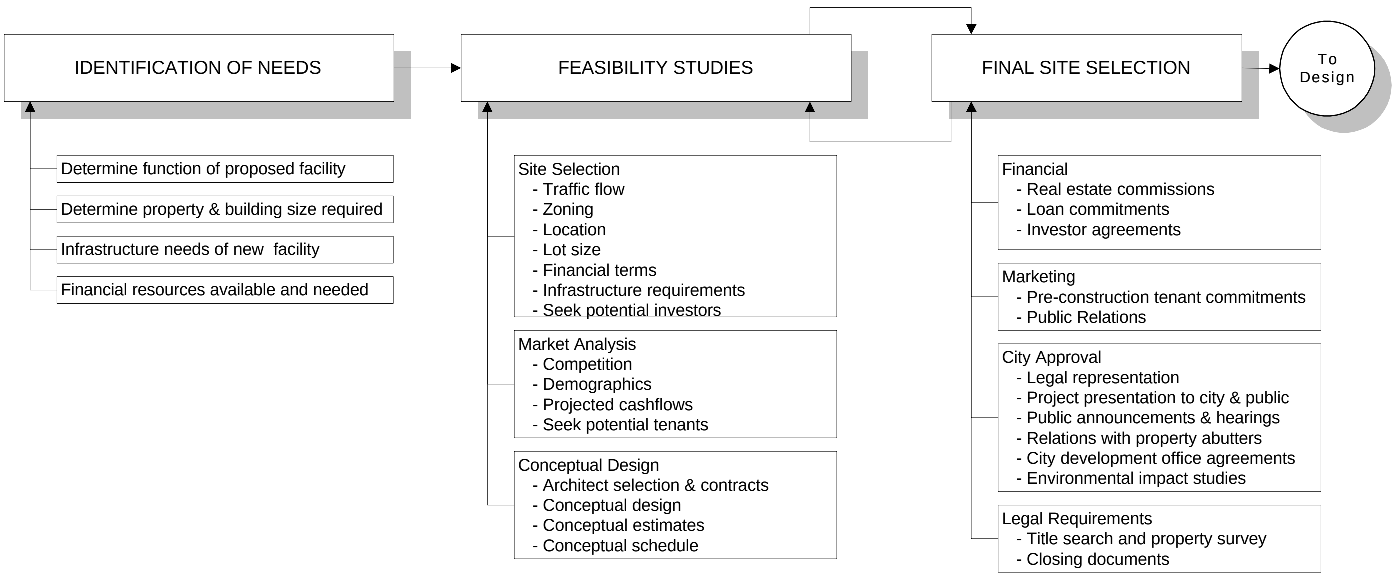
2.1. General Phases of a Building Project

The diagrams below represent the traditional AEC/FM processes where the four phases are represented as linear processes accomplished over time even though cycles exist within the phases and between phases. Each of the phases has a discipline which is responsible contractually for the completion of the phase. Disciplines may span across the phase but usually their input represents overseeing previous work for which they were responsible. Each of the following phases is described to provide the reader with a brief background.



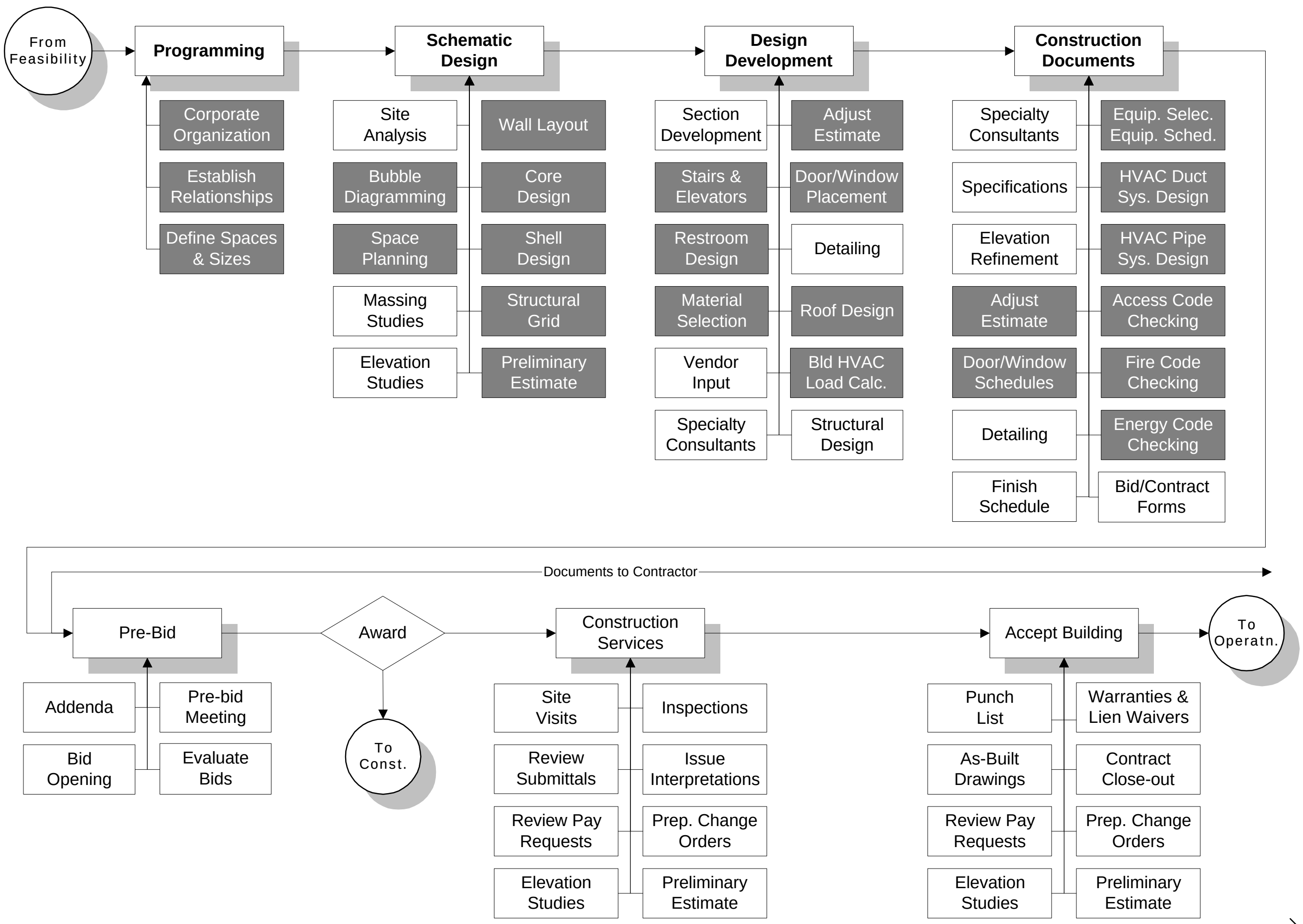
2.1.1. Feasibility Phase

The Feasibility Phase involves the need to expand or re-arrange a facility or facilities. The process involves defining the best method for the building owner, developer, or corporation to fill their long term need for space. The decision may be between renovating an existing building, leasing space, building a new facility, or any combination of the above. At this point, a program is created by the client, facility programming consultant, or an architect to determine the capacity of the facility. Other related issues are researched that may impact the project both legally or financially.



2.1.2. Design Phase

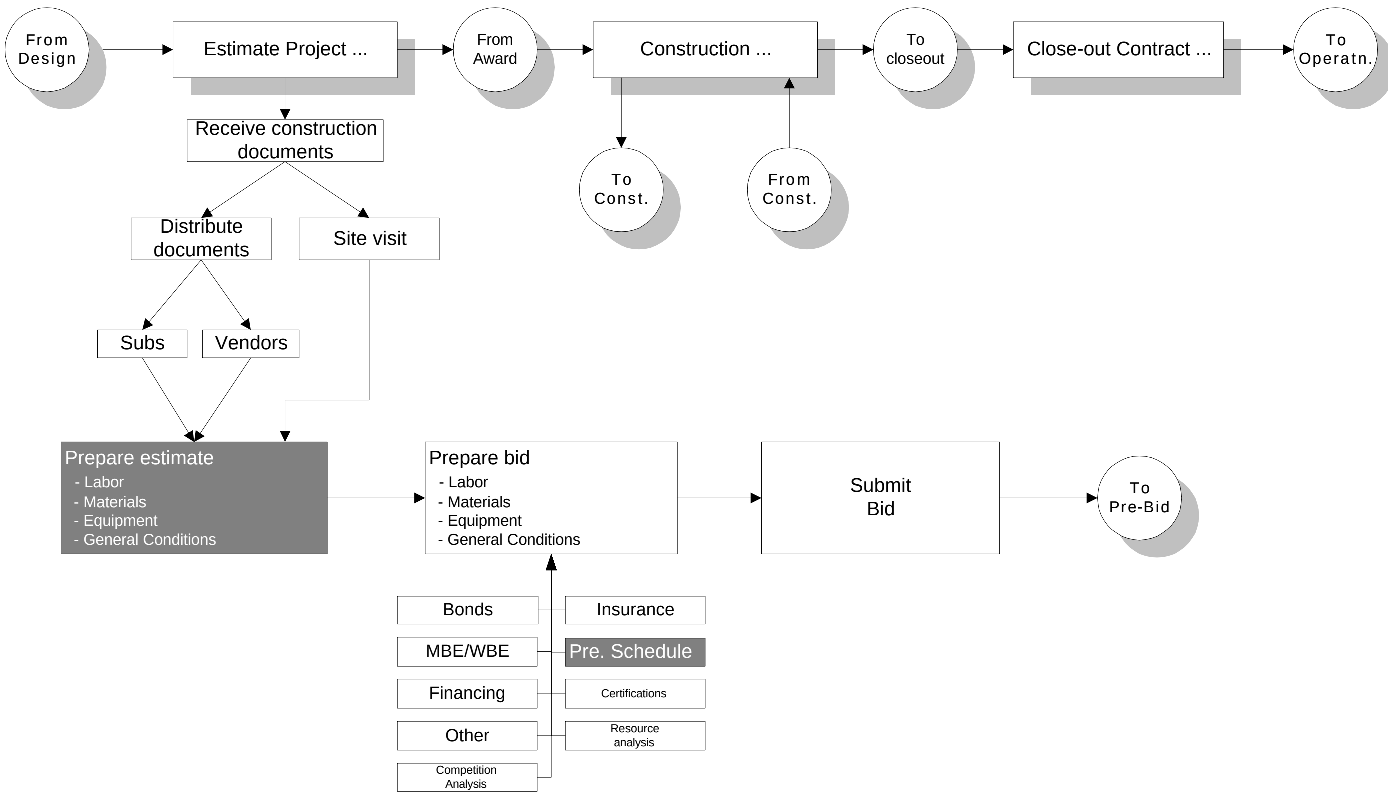
Depending on the decision in the previous Feasibility Phase, the Design Phase may range from just the interior layout and design of existing space up to the design of a new facility using the full range of disciplines, ie. architecture, interior, engineering, and specialty consultants. The traditional project has the architect, through contracts with the client, responsible for the final product of this phase, which is a set of drawing and specifications in electronic or paper format. The drawings provided by the rest of the team in this process such as the engineers are rolled up with the architects as a single set of information for the construction of the facility.



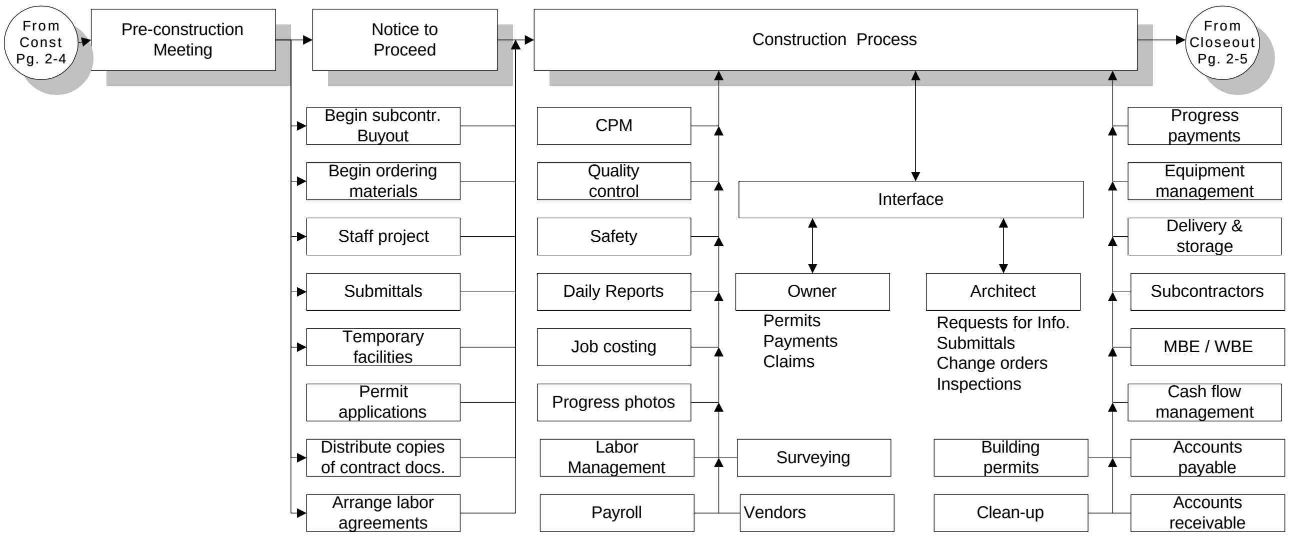
2.1.3. Construction Phase

The Construction Phase begins with an analysis of the drawings and specification documents to create an estimate on time, equipment, material, and manpower needed to construct the facility. The contractor is selected usually as part of a Bid Process where competing estimates allow the client to determine the appropriate company to build the project. A construction manager may be hired by the client to oversee the transition between the design phase and construction phase and provide scheduling and financial management for the project.

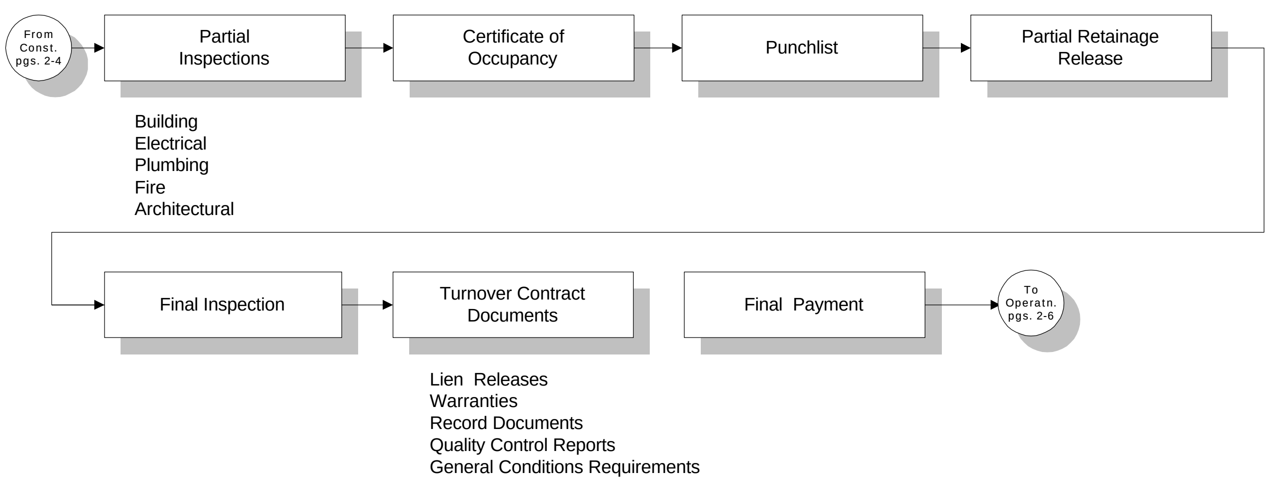
2.1.3.1. Project Estimation Processes



2.1.3.2. Construction Process

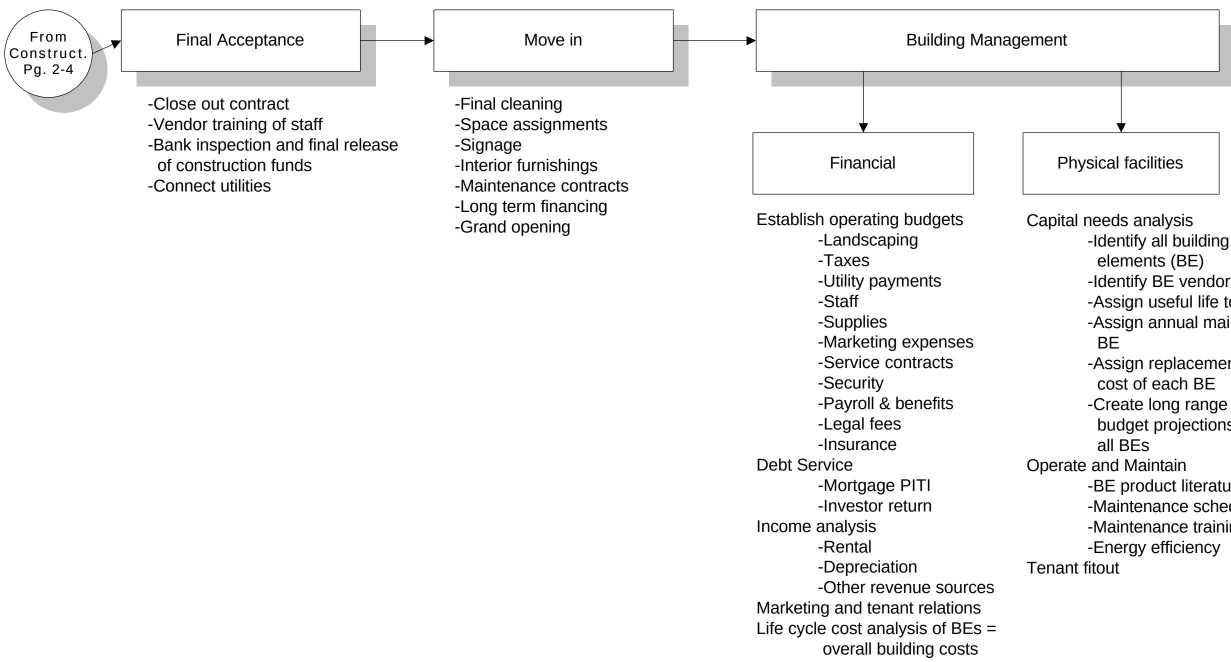


2.1.3.3. Close-Out Processes



2.1.4. Operation Phase

By far the Operation Phase is the longest and most expensive of the four phases outlined. The Operation Phase begins after the contractor turns over a finished building and the client receives an occupancy permit that indicate that the facility is inhabitable. The operation phase has many cycles and involves a large array of specialties to manage, track, operate, and maintain the facility through its continued life.

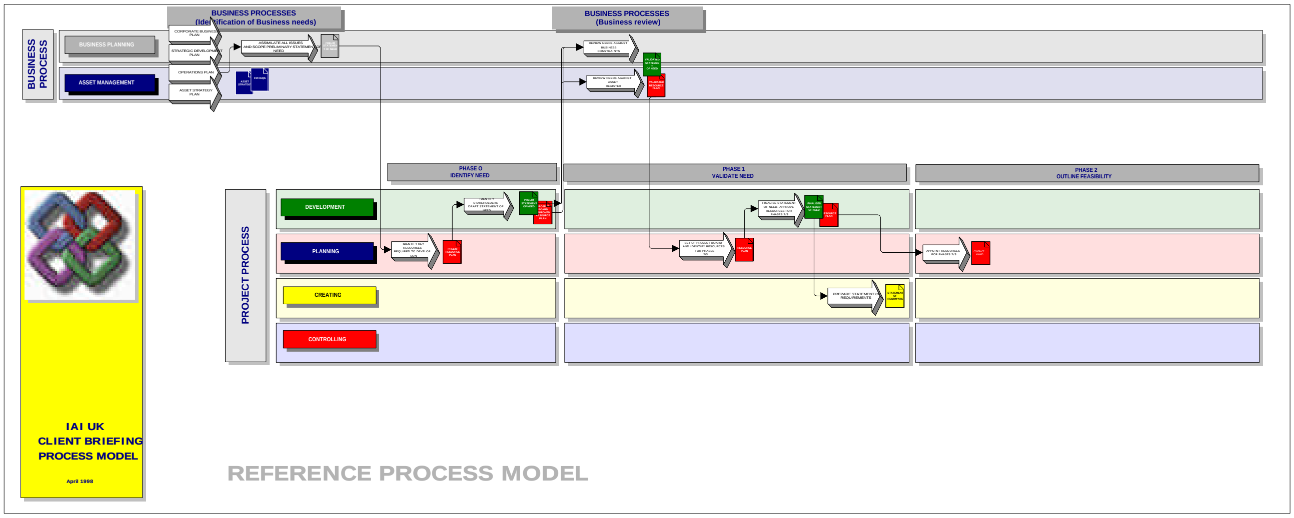


1.1. Work in process - Reference AEC/FM Process Model

The IAI's CB-1 team was formed in 1997. Their charter states they mission as "specify data required at each stage of the project process, and the operation of the building from the client viewpoint. This information should then form part of the Brief to each of the other domain groups as they determine object specifications." The process framework in the previous section will be replaced by the model developed by this team. This section gives you a glimpse of the approach being taken.

2.1.5. General Phases of a Building Project

The diagram below indicates how a building project is developed out of a building owners business processes. It shows the overall framework in which business and building processes take place. A key feature of this model is to illustrate how activities in a building project can be seen being delivered out of generic concurrent processes.



The generic business processes are ' Business Planning' and 'Facilities Management' . These take place concurrently with any development project (s). The lower part of the model comprises the building process model and comprises 'Development'; 'Planning'; 'Implementing' and Controlling processes. The building process model also comprises of key stages. These are described below:

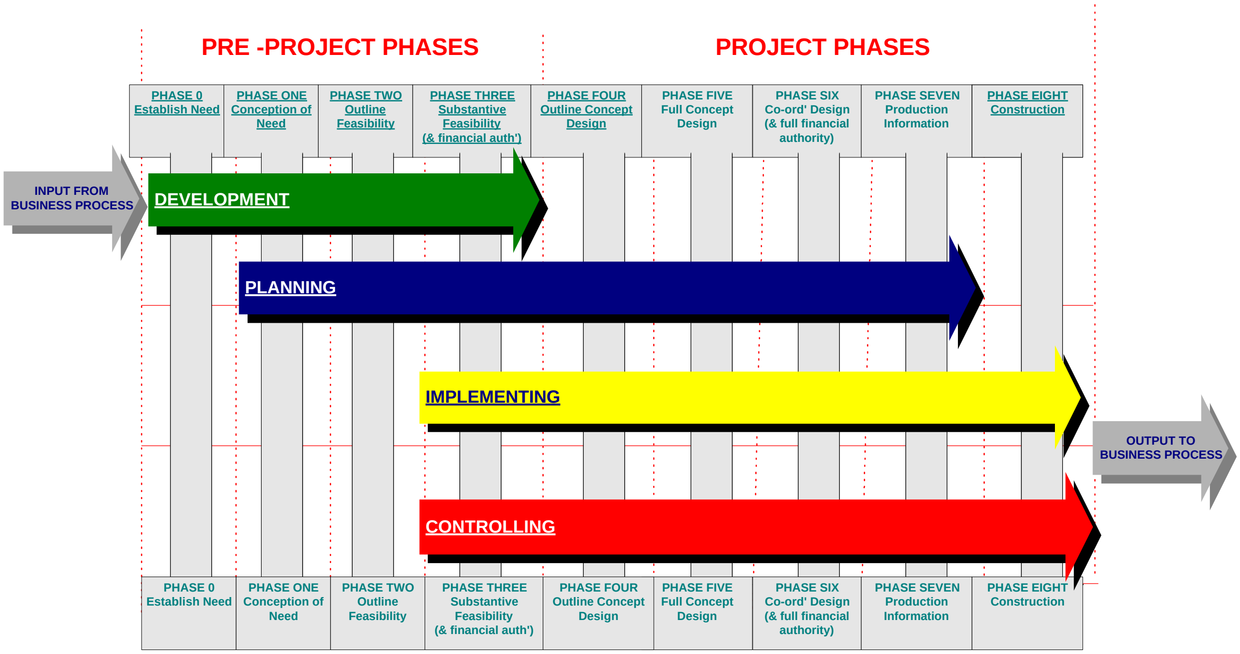
PRE -PROJECT PHASES

PROJECT PHASES

PHASE 0 Establish Need	PHASE ONE Conception of Need	PHASE TWO Outline Feasibility	PHASE THREE Substantive Feasibility (& financial auth')	PHASE FOUR Outline Concept Design	PHASE FIVE Full Concept Design	PHASE SIX Co-ord' Design (& full financial authority)	PHASE SEVEN Production Information	PHASE EIGHT Construction

The pre-project phases run from Phase 0 to Phase 4. During these phases the Brief for the project is determined and sufficient design work is carried out to ensure that a robust business case, with sufficient briefing and design information is produced. This will enable a project team to deliver the project to targets set by the building owner and agreed at the end of phase 3.

During Phases 4 to 8 the team is responsible for both the on-going brief development; design and construction. The processes ensure that the right skills are available in the team at each stage to meet the demands of the project. In Phase 3 the initial product data model is developed (although this could happen earlier) and the model is then enhanced with more attribute information to produce the electronic prototype of the proposed facility. During stages 5 and 6, in particular, simulation technologies are used to optimise the design to ensure the closest possible fit with the Brief. During Phase 7, the construction process is simulated to ensure that a properly informed and optimised plan is established before assembly and construction take place. In Phase 8, construction takes place as planned and the facility is handed over in accordance with the building project plan.



REFERENCE PROCESS MODEL FROM CLIENT PERSPECTIVE

All the work of the team within a building project should be customer focussed. The business process model articulates customer-focussed measures. It is these measures that every process and sub-process should be focussed on achieving.

The key processes in the are illustrated in the above diagram.

Development process. This is the process that receives the output from the Business Planning process, where customerst needs are first identified. The Development Process, analyses the Statement of Need and should the process identify that the need will be best met through a building project, then resources are identified and a project Brief called a Statement of Requirements. Concurrent with this process is a Planning Process where an overall strategy for achieving the building project is developed. The Planning Process is where all the detailed plans for delivering the project to meet the development requirements are formulated.

It is in the Implementation Process, where the detailed Briefing, Design (and where the development of Product Data Model is produced). It is where Procurement and Construction take place, as the building project evolves from a design solution to constructed reality.

Concurrent with the Implementing process is the Controlling Process. This is where controls are established to ensure that the building project proceeds in accordance with the planned process. It is here where outputs from the Development Process and the Implementation Process are reviewed against the Brief and the various plans to ensure conformance with requirements. It is here too, where change is controlled and the implications of change on the building project are understood.

A key implication of concurrent processes is the need to manage information exchanges in a much more controlled manner than with linear processes. Object technology is a key enabler to effective information management. In developing processes and sub-processes the information exchanges are modelled and the attribute data for objects is defined. As objects inherit attribute data they 'carry' it from one process to the next, and so facilitate the information exchanges.

3. Requirements Definition Project Summaries

3.1. [AR-1] Architectural Model Extensions

3.1.1. Project Description

AEC Industry Processes described in this project:

- Shell Design
- Core Design
- Stair Design
- Restroom Design
- Roof Design

This project will define these five processes in an effort to complete the basic Architectural Model for a commercial office building. These processes span from the Schematic design phase of Architecture through refinement in the Construction Document phase.

3.1.2. Project Team

Project Leader ***Ken Herold - North America –*** Ken.Herold@hok.com

Chap	Name	Company	Phone	Email	Hrs / Wk
NA	Gustavo A. Lima	Cannon		glima@cunnon.com	?
	Bill O'Malley	Hammel Green and Abrah...		BOMalley@EMAIL.HGA.COM	?
	Barbara Golte...	Heller & Metzger, PC		74212.354@compuserve.com	?
	Ken Herold	HOK		iaiexec@interoperability.com	5
	Steve Stevens	Intergraph		festeven@ingr.com	3
	Juniper Russell	Juniper Russell & Assoc.		juniper@novimundi.com	?
	Ed Ebbing	MC2		eebbing@mc2-ice.com	1
	Martin Rozmanith	RMW Architecture + Design		marty_rozmanith@rmw.com	?
	Ardie Aliandust	RTKL		2350@la.rtkl.com	3
	Bill Houstoni	RTKL		bhouston@balt.rtkl.com	?
	Nick Reveliotty	The Kling Lindquest Pa...			?
	Tony Sinisi	The Kling Lindquest Pa...		76636.1043@compuserve.com	?
	Beth Brucker	USA-CERL		B-Brucker@cecer.army.mil	2
	Paul Lewis	Visio		paull@visio.com	?
	Rob Wakeling	Visio		robw@visio.com	?
				Team total =	**14**

3.1.3. Scope of Work

# of AEC processes to be supported	-	6	Est. total AEC expert time (days)	-	30.2
Expected IFC Model Impact (1 *(min)* to 5)	-	4	Est. total Info Modeling expert time (days)	-	??
Degree of technical difficulty (1 *(min)* to 5)	-	4	Est. total Project Mgmt. expert time (days)	-	??

3.1.4. Resources Required / Committed

Member Company Resources	*Reqired Days*	*Market Value*		*Days Committed*	*Resource shortfall*
Process Model	30	$10,195		nn	nn
Usage Requirements	30	$10,195		nn	nn

Object Model development	30	$10,195		nn	nn
Integration	7.5	$2,600		nn	nn
Test Case development	37.75	$12,740		nn	nn
Implementation technical support	7.5	$2,600		nn	nn
Management and Review	7.5	$2,600		nn	nn
Total Member Company Resources	**151**	**$51,000**		nn	nn
Travel		**$68,000**			
Project Support	**Required Days**	**Market Value**			
Technical support	nn	$nn			
Project management	nn	$nn			
Publication and Administration	nn	$nn			
Equipment and software	nn	$nn			
Travel and subsistence	nn	$nn			
Total Project Support	**nn**	**$nn**			
Total for Project	**nn**	**$nn**			

3.2. [AR-2] Compartmentation of Buildings

3.2.1. Project Description

AEC Industry Processes described in this project:

- Compartmentation of Building

It is assumed that the fire usually starts in one place, and spreads to other parts of the building. In order to allow the occupants of the building to escape, the first thing is to stop the spread of fire and smoke to other parts of the building, as well as to maintain common escape routes free of fire and smoke. Compartmentation allows the control of fire within a limited space, allowing occupants of the building to escape and to control the fire.

On receipt of Architects drawings, identify primary and secondary Use Classes, for the total project. In doing so define shape and size of each Use Class compartments will be defined. Use Class compartments [Proposed Compartments] may need to be sub-divided into Occupancy type [Owners/Tenants] compartments and if necessary sub-divided further to meet the maximum permitted floor area or the volume for a given compartment. Final result is the Fire compartment.

3.2.2. Project Team

Project Leader ***Jay Patankar -*** *patankar@dial.pipex.com*

Chap	***Company***	***Member***	***Phone***	***Email***	***Hrs / Wk***
UK		Jay Patankar		patankar@dial.pipex.com	
UK		Steve Race		darcyrace@dial.pipex.com	
UK		John Cann			
UK		David Clarke		David.clarke@eur.autodesk.com	
UK		Jeffrey Wix		100125.2426@compuserve.com	
UK		Donald Ross		ssi@ltd.net	
UK		Richard Harpham		richard_harpham@compuserve.com	

3.2.3. Scope of Work

# of AEC processes to be supported	-	3	Est. total AEC expert time (days)	-	30
Expected IFC Model Impact (1 *(min)* to 5)	-	5	Est. total Info Modeling expert time (days)	-	40
Degree of technical difficulty (1 *(min)* to 5)	-	4	Est. total Project Mgmt. expert time (days)	-	??

3.2.4. Resources Required / Committed

Member Company Resources	**Reqired Days**	**Market Value**		**Days Committed**	**Resource shortfall**
Requirements definition					
Process Model	30	£13200			
Usage Scenaria	20	£8800			
Model design					
Object Model development *(w/ tech.Support)*	40	£17600			
Integration *(w/ tech.Support)*	10	£4400			
Design and Implementation validation					
Test Case development	20	£8800			
Review/feedback on implementations	40	£17600			
Project Management					
Project management and administration	34	£14960			
Travel and Meetings	80	£35200			
Total Member Company Resources	**274**	**120560**			

Model/Specification development support	**Required Days**	**Market Value**			
Technical support	50	£12500			
Project management	24	£10500			
Publication and Administration	10	£2200			
Equipment and software		£2000			
Travel and subsistence		£2000			
Total Project Support		**£29200**			
Total for Project		**£149760**			

3.3. [BS-1] HVAC System Design

3.3.1. Project Description

AEC Industry Processes described in this project:

- HVAC Duct Design
- HVAC Piping Design

These processes will involve utilizing the network object types defined in the IFC 2.0 Core model. This effort will be led by the North American Building Services Committee, but will be an international collaborative effort. This will ensure that the resulting system design extensions are globally applicable.

Engineers responsible for the design of duct and piping systems may be consulted during the building conceptual stage. However, the major design effort occurs after the architect has substantially completed the building drawings. The design process includes both the schematic and detailed description of duct and piping components. These components include sections of duct and pipe, fittings, accessories such as dampers, valves, and terminals. This process also includes the connection of these components to equipment such as fans and pumps. Object types for equipment were defined in IFC Version 1.x, and are not elaborated in this proposal. The system design process also includes construction cost estimates. However, these estimates are typically performed by contractors using the drawings and specifications prepared by the Building Services engineer.

Significant cost savings will result from the application of IFC's to systems design in Building Services.

- Building geometry and construction materials used in the design of HVAC load calculations and the fluid distribution systems.
- The exchange of data between engineering design and analysis programs with manufacturers' equipment selection programs.
- The production of schedules of bill of materials for the system components.
- Producing the data for engineers cost estimates and for contractors actual construction cost estimates.
- The opportunity for integration of control components used for the operation of these systems.

3.3.2. Project Team

Project Leader ***James Forester -*** *jim@marinsoft.com*

Chap	Company	Name	Phone	Email	Hrs / Wk
NA		John Deal		75601.1346@compuserve.com	4
NA		Rod Dougherty		rod.dougherty@landis+gyr.sprint.com	4
NA		James Forester		jim@marinsoft.com	4
NA		Scott Frank		sfrank@pipeline.com	2
NA		Kirk McGraw		k-mcgraw@cecer.army.mil	2
NA		Larry Schaefer		larry.schaefer@carrier.wltk.com	2
NA		Tony Sherfinski		tony.sherfinski@greenheck.com	2
UK		Jeff Wix		100342.2537@compuserve.com	2
				Total for project team =	***22***

3.3.3. Scope of Work

# of AEC processes to be supported	-	2	Est. total AEC expert time (days)	-	40
Expected IFC Model Impact (1 *(min)* to 5)	-	3	Est. total Info Modeling expert time (days)	-	40
Degree of technical difficulty (1 *(min)* to 5)	-	3	Est. total Project Mgmt. expert time (days)	-	40

3.3.4. Resources Required / Committed

Member Company Resources	Required Days	Market Value		Days Committed	Resource shortfall
Requirements definition					
Process Model	10	$12000		10	0
Usage Scenaria	15	$18000		15	0
Model design					

Object Model development *(w/ tech.Support)*	10	$12000		5	5
Integration *(w/ tech.Support)*	20	$24000		10	10
Design and Implementation validation					
Test Case development	15	$18000		10	5
Review/feedback on implementations	15	$18000		??	??
Project Management					
Project management and administration	15	$18000		30	nn
Travel and Meetings	10	$12000		10	nn
Total Member Company Resources	**110**	**$132000**		85+	40+

Model/Specification development support	**Required Days**	**Market Value**		**Days Committed**	**Resource shortfall**
Technical support	5	$6000			
Project management	10	$12000			
Publication and Administration	10	$12000			
Equipment and software	5	$6000			
Travel and subsistence	10	$12000			
Total Project Support	**40**	**$48000**			
Total for Project	**150**	**$180000**			

3.4. [BS-3] Pathway Design and Coordination

3.4.1. Project Description

AEC Industry Processes described in this project:

- Pathway Design and Coordination

The design of pathways contains the draft layout, the coordination and the representation of mechanical and electrical system-pathways to be installed.

This design process is carried out after the first coordination with the architect and structural engineers, and includes load estimates, energy and systems definitions required for a building.

The process ends with drawings containing the coordinated pathways for the mechanical and electrical installations (i.e. heating, cooling, air-conditioning, plumbing, fire-protection and electrical power) within a building.

3.4.2. Project Team

Project Leader   Rolf Tonke / Bertram Witz - German Chapter

Chap	Company	Member	Phone	Email	Hrs
Germany	vögtlin engineering	Felix Brückner		100737.1421@compuserve.com	0
Austria	PHi-Tech	Bernhard Fragner		fragner@phitech.co.at	0
Germany	GTS	Rainer Hirschberg		Rh@gts-software.com	40
Austria	‘ESS	Doris Huber		ess@klima2000.co.at	0

Germany	Softtech	Eberhard Michaelis		EMichaelis@softtech.com	80
Germany	Ziegler Informatics	Ulrich Paar		ziegler@caddy.de	0
Swiss	RoCAD Informatik	Robert Rottermann		100041.2347@compuserve.com	80
Germany	Triplan GmbH	Willi Spiegel		willi.spiegel@triplan.com	0
Germany	Planungsgruppe M+M AG	Rolf Tonke		100436.705@compuserve.com	120
Germany	Pit-cup GMBH	Kurt Weber		pit-cup@t-online.de	0
Germany	Planungsgruppe M+M AG	Bertram Witz			80
Germany	Kuehn Bauer Partner	Michael Kuehn jr.		mkj@kbp-futures.com	0
				Total for project team	***400***

3.4.3. Scope of Work

# of AEC processes to be supported	-	7	Est. total AEC expert time (days)	-	18
Expected IFC Model Impact (1 *(min)* to 5)	-	3	Est. total Info Modeling expert time (days)	-	15
Degree of technical difficulty (1 *(min)* to 5)	-	2	Est. total Project Mgmt. expert time (days)	-	15

3.4.4. Resources Required / Committed

Member Company Resources	**Required Days**	**Market Value**		**Days Committed**	**Resource shortfall**
Requirements definition					
Process Model	10	$7060		nn	nn
Usage Scenaria	7	$4942		nn	nn
Model design					
Object Model development *(w/ tech.Support)*	5	$3530		nn	nn
Integration *(w/ tech.Support)*	8	$5648		nn	nn
Design and Implementation validation					
Test Case development	10	$7060		nn	nn
Review/feedback on implementations	5	$3530		nn	nn
Project Management					
Project management and administration	2	$1412		nn	nn
Travel and Meetings	3	$2118		nn	nn
Total Member Company Resources	**50**	**$35300**		nn	nn

Model/Specification development support	**Required Days**	**Market Value**		**Days Committed**	**Resource shortfall**
Technical support	nn	$nn			
Project management	nn	$nn			
Publication and Administration	nn	$nn			
Equipment and software	nn	$nn			
Travel and subsistence	nn	$nn			
Total Project Support	**nn**	**$nn**			
Total for Project	**nn**	**$nn**			

3.5. [BS-4] HVAC Loads Calculation

3.5.1. Project Description

AEC Industry Processes described in this project:

- Building Heating and Cooling Load Calculation

Load calculations serve as the basis for all design stages of the building services design. The results of the load calculations enable the designer to dimension the plant equipment and to determine the required space for plant room.

Load calculations are an official proofing method in Germany for example the proof for heat loss protection must be given in the course of a project), a mode for calculating the heating cooling load or for the yearly dynamic load simulation:

The process terminates in the complete calculations and the data exchange into the IFC model.

3.5.2. Project Team

Project Leader Rolf Tonke / Rainer Hirschberg - German Chapter

Chap	Company	Member	Phone	Email	Hrs
Germany	vögtlin engineering	Felix Brückner		100737.1421@compuserve.com	0
Austria	PHi-Tech	Bernhard Fragner		fragner@phitech.co.at	0
Germany	GTS	Rainer Hirschberg		Rh@gts-software.com	140
Austria	'ESS	Doris Huber		ess@klima2000.co.at	0
Germany	Softtech	Eberhard Michaelis		EMichaelis@softtech.com	140
Germany	Ziegler Informatics	Ulrich Paar		ziegler@caddy.de	0
Swiss	RoCAD Informatik	Robert Rottermann		100041.2347@compuserve.com	80
Germany	Triplan GmbH	Willi Spiegel		willi.spiegel@triplan.com	0
Germany	Planungsgruppe M+M AG	Rolf Tonke		100436.705@compuserve.com	160
Germany	Pit-cup GMBH	Kurt Weber		pit-cup@t-online.de	0
Germany	Planungsgruppe M+M AG	Bertram Witz			80
Germany	Kuehn Bauer Partner	Michael Kuehn jr.		mkj@kbp-futures.com	40
				Total for project team =	640

3.5.3. Scope of Work

# of AEC processes to be supported	-	6	Est. total AEC expert time (days)	-	30
Expected IFC Model Impact (1 *(min)* to 5)	-	5	Est. total Info Modeling expert time (days)	-	20
Degree of technical difficulty (1 *(min)* to 5)	-	2	Est. total Project Mgmt. expert time (days)	-	20

3.5.4. Resources Required / Committed

Member Company Resources	**Reqired Days**	**Market Value**		**Days Committed**	**Resource shortfall**
Requirements definition					
Process Model	14	$9884		nn	nn
Usage Scenaria	12	$8472		nn	nn
Model design					
Object Model development *(w/ tech.Support)*	5	$3530		nn	nn
Integration *(w/ tech.Support)*	8	$5648		nn	nn
Design and Implementation validation					

Test Case development	15	$10590		nn	nn
Review/feedback on implementations	8	$5648		nn	nn
Project Management					
Project management and administration	5	$3530		nn	nn
Travel and Meetings	5	$3530		nn	nn
Total Member Company Resources	**72**	**$50832**		nn	nn

Model/Specification development support	**Required Days**	**Market Value**		**Days Committed**	**Resource shortfall**
Technical support	nn	$nn			
Project management	nn	$nn			
Publication and Administration	nn	$nn			
Equipment and software	nn	$nn			
Travel and subsistence	nn	$nn			
Total Project Support	**nn**	**$nn**			
Total for Project	**nn**	**$nn**			

3.6. [CS-1] Code Checking - Energy Codes

3.6.1. Project Description

AEC Industry Processes described in this project:

- Commercial and Residential Energy Code Compliance Checking

This project has two parts: CS-1A - ***Code Compliance Enabling Mechanism*** and CS-1B ***Energy Code Compliance Checking***. These two parts have been combined into a single project for administrative efficiency. Part A of the project will define a generic code compliance enabling mechanism that will be applicable to codes of various types; e.g., accessibility, egress, and energy. The mechanism will likely involve defining new abstract classes for code compliance in the core model. Part A will be an international collaborative effort, which will ensure that the resulting enabling mechanism is broadly applicable. Part B, Energy Code Compliance, will serve an important role in validation of the generic mechanism for a set of code applications. This work will be performed primarily by the North American Chapter and will enable established energy code compliance applications to be made IFC compliant.

Code compliance checking is performed by building designers, systems designers, and code enforcement officials. Compliance with codes begins during the programming phase when designers determine which codes apply to the building project. Preliminary code reviews are frequently performed during schematic design, and more thorough reviews are performed by members of the design team late in the design process before construction documents are complete. Building code officials perform plan reviews as part of the building permit process. Designers and code official perform drawing dimension takeoffs as necessary to ensure compliance. Information about building systems, assemblies, layout, etc. is gathered during this process and compared to the requirements for each applicable code.

Codes impact virtually all disciplines involved in building design and construction processes, and code considerations persist throughout a building's life cycle. Energy codes are strongly related to architectural, HVAC, and electrical design processes. While it would be difficult to establish a reliable estimate of time and cost savings from IFC support of code checking, the tedious nature of code review and the large cost and schedule impacts that code violations can cause suggest that there will be high demand for code checking applications. Energy codes represent an attractive application for IFC support because of their extensive requirements for building data that are already in electronic form (e.g., geometric data and lighting fixture data) and demonstrated strong demand--thousands of copies of these applications currently in use.

3.6.2. Project Team

Project Leader Rob Briggs - North America Chapter

Chap	Company	Member	Phone	Email	Hrs / Wk
N. America		Rob Briggs		rs_briggs@pnl.gov	10
Singapore		Tan You Tong		youtong@iti.gov.sg	2
France		Philippe Debras		debras@cstb.fr	2
UK		Robert Amor		trebor@bre.co.uk	1
				Total for project team =	15

3.6.3. Scope of Work

# of AEC processes to be supported	-	1	Est. total AEC expert time (days)	-	5
Expected IFC Model Impact (1 *(min)* to 5)	-	2	Est. total Info Modeling expert time (days)	-	2
Degree of technical difficulty (1 *(min)* to 5)	-	3	Est. total Project Mgmt. expert time (days)	-	2

3.6.4. Resources Required / Committed

Member Company Resources	*Reqired Days*	*Market Value*		*Days Committed*	*Resource shortfall*
Requirements definition					
Process Model	5	$4,000		5	0
Usage Scenaria	5	$4,000		5	0
Model design					
Object Model development *(w/ tech. Support)*	10	$8,000		10	0
Integration *(w/ tech. Support)*	8	$6,400		8	0
Design and Implementation validation					
Test Case development	5	$4,000		5	0
Review/feedback on implementations	5	$3,840		5	0
Project Management					
Project management and administration	5	$4,000		5	0
Travel and Meetings	5	$7,000		5	0
Total Member Company Resources	**48**	**$41,240**		48	0

Model/Specification development support	**Required Days**	**Market Value**		**Days Committed**	**Resource shortfall**
Technical support	nn	$nn			
Project management	nn	$nn			
Publication and Administration	nn	$nn			
Equipment and software	nn	$nn			
Travel and subsistence	nn	$nn			
Total Project Support	**nn**	**$nn**			

Total for Project	**nn**	**$nn**			

3.7. [CS-2] Code Checking Extensions

3.7.1. Project Description

AEC Industry Processes described in this project:

- Code Compliance - Disabled Access
- Code Compliance - Escape Routes

The project covers specific application of the code compliance enabling mechanism (R2_CS-1) in serving the disabled access and escape routes code compliance.

Disable access code compliance is a process of assessing whether **the access provisions and facilities** of a building complies with one or more codes or standards **that serve the needs of the wheelchair user and ambulant disabled** enforced by various codes and standards promulgation entities.

Escape route code compliance is a process of assessing whether **the exit provisions and facilities** of a building complies with one or more codes or standards **that provide safe means of escape for occupants** enforced by various codes and standards promulgation entities.

The processes are performed by building designers and code enforcement officials during early design and submission stages, respectively. Automatic code compliance software based on the IFC models created in this project will help building designers to carry out self-checking of their designs in order to detect code violations as early as possible while design changes are still relatively cheap to make. Similarly, it also help the code enforcement officials to verify the plans submitted by the designers for building approvals.

The resources required to produce the IFC model for the disabled access and escape route are estimated to be 160 man-days over 20 elapse calendar weeks. Based on market value of $200 (Singapore) per man-days, a total of $32000 is required for the project.

3.7.2. Project Team

Project Leader ***Mr. Wong Wai Ching - Singapore***

Disable Access

Chapter	***Name***	***Email***	***Hrs / Week***
Singapore	Mr. Wong Wai Ching (leader)	- keewee@ncb.gov.sg	2
Singapore	Mr. Zhong Qi (info modeling)	Zhongqi@ncb.gov.sg	22
Singapore	Mr. Liew Pak San (software)	paksan@ncs.com.sg	4
Singapore	Dr. Tan Kee Wee	Keewee@ncb.gov.sg	4

A total of 32 man-hrs/week is required which is equivalent to 4 man-days/week (based on 8 hrs/days). Over 20 calender weeks, a total of 80 man-days is required.

Escape Route

Chapter	***Name***	***Email***	***Hrs / Week***
Singapore	Mr. Wong Wai Ching (leader)	- keewee@ncb.gov.sg	2
Singapore	Mr. Zhong Qi (info modeling)	Zhongqi@ncb.gov.sg	22
Singapore	Mr. Liew Pak San (software)	paksan@ncs.com.sg	4

Singapore	Dr. Tan Kee Wee	Keewee@ncb.gov.sg	4

A total of 32 man-hrs/week is required which is equivalent to 4 man-days/week (based on 8 hrs/days). Over 20 calender weeks, a total of 80 man-days is required.

3.7.3. Scope of Work

AEC processes to be supported	-	2	Est. total AEC expert time (weeks)	-	2
Expected IFC Model Impact (1 *(min)* to 5)	-	3	Est. total Info Modeling expert time (weeks)	-	10
Degree of technical difficulty (1 *(min)* to 5)	-	3	Est. total Software/PM expert time (weeks)	-	4

3.7.4. Resources Required / Committed

Member Company Resources	**Reqired Days**	**Market Value**		**Days Committed**	**Resource shortfall**
Process Model	4	S$200		4	0
Usage Requirements	20	S$200		20	0
Object Model Development	80	S$200		80	0
Integration	20	S$200		20	0
Test Case Development	10	S$200		10	0
Implementation Technical Support	10	S$200		10	0
Management and Review	16	S$200		16	0
Total Member Company Resources	**160**	**S$32000**		**160**	**0**
Project Support	**Required Days**	**Market Value**			
Technical support	30	$15000			
Project management	5	$2500			
Publication and Administration	-	$500			
Equipment and software	-	$1000			
Travel and subsistence	0	$0			
Total Project Support	**35**	**$19000**			
Total for Project	**195**	**$51000**			

3.8. [ES-1] Cost Estimating

3.8.1. Project Description

AEC Industry Processes described in this project:

- Code Compliance - Disabled Access

This project is designed to increase the ability of the model to support cost estimating. The model already supports cost estimating to some degree. This project focuses refining and expanding that capability.

Most of the information used by cost estimating will be entered into the model by earlier design processes. At various times during the evolution of the design, an estimator will use the model to do cost estimating. During early design stages, very little information will be available, and only a rough estimate will be possible. As the model becomes more detailed, more accurate estimates are possible. When different designs are under consideration, “what if” or “alternate” estimates may be used to compare their cost impact. After a design and estimate are approved, inevitably, changes will be proposed and “change order” estimates will be required to determine the cost impact of the proposed change.

Using the IFC Model to do cost estimating saves time by using information provided by the design processes. It can also save time by making the task and resource data that it creates available to later processes such as scheduling. Using the model as the primary information source for estimating can also reduce errors and omissions that occur when data is entered into an estimating system by hand.

3.8.2. Project Team

Project Leader ***Mike Cole - North American Chapter***

Chapter	Name	Email	Hrs / Wk
NA	Mike Cole	mikec@timberline.com	10
NA	Ray Brungard	rbrungard@tcco.com	.5
UK	Jeffrey Wix	10342.2537@compuserve.com	?
NA	Peggy Woodall	peggy@bsdsoftlink.com	?
NA	Annette Stumph	a-stumpf@cecer.army.mil	?
NA	Roger Grant	rgrant@rsmeans.com	?
DE	Hans-Peter Sanio	San@mail.rib.de	?
Total for project			

3.8.3. Scope of Work

AEC processes to be supported	-	1	Est. total AEC expert time (days)	-	nn
Expected IFC Model Impact (1 *(min)* to 5)	-	1	Est. total Info Modeling expert time (days)	-	nn
Degree of technical difficulty (1 *(min)* to 5)	-	1	Est. total Software/PM expert time (days)	-	nn

3.8.4. Resources Required / Committed

Member Company Resources	*Reqired Days*	*Market Value*		*Days Committed*	*Resource shortfall*
Requirements definition					
Process Model	2	$1000		2	0
Usage Scenaria	4	$2000		4	0
Model design					
Object Model development *(w/ tech.Support)*	10	$5000		10	0
Integration *(w/ tech.Support)*	6	$3000		6	0
Design and Implementation validation					
Test Case development	6	$3000		6	0
Review/feedback on implementations	8	$4000		8	0
Project Management					
Project management and administration	8	$4000		8	0
Travel and Meetings	50	$25000		50	0
Total Member Company Resources	**94**	**$47000**		86	

Model/Specification development support	**Required Days**	**Market Value**		**Days Committed**	**Resource shortfall**
Technical support	nn	$nn			
Project management	nn	$nn			
Publication and Administration	nn	$nn			
Equipment and software	nn	$nn			
Travel and subsistence	nn	$nn			
Total Project Support	**nn**	**$nn**			
Total for Project	**nn**	**$nn**			

3.9. [FM-3] Property management (Building Owner's viewpoint)

3.9.1. Project Description

Property management is a process starting from requirement programming and continuing through the building's life cycle. In this phase the FM-3 project covers just a subset of this process focusing on grouping of spaces and other possible objects for different purposes, like maintenance, administration, public registers, mapping etc. This process is based on objects provided by the design and construction process and uses mainly the attributes in the current model. The main user is the building owner and the benefit is more efficient use of the building data and through this cost savings in the administrative work. This process starts after the building is completed and is carried out through the whole life cycle of the building.

3.9.2. Project Team

Project Leader ***Poul Sorgenfri Ottosen - Nordic Chapter***

Chapter	Name	Email	Hrs / Wk
Nordic	Poul Sorgenfri Ottosen	pso@aua.auc.dk	10
Nordic	Jan Karlshøj	jakbyg@carlbro.dk	3
Nordic	Arto Kiviniemi	arto.kiviniemi@vtt.fi	5
NA	Kevin Yu		1
		Total for project team =	***19***

3.9.3. Scope of Work

AEC processes to be supported	-	3	Est. total AEC expert time (days)	-	30
Expected IFC Model Impact (1 *(min)* to 5)	-	1	Est. total Info Modeling expert time (days)	-	15
Degree of technical difficulty (1 *(min)* to 5)	-	1	Est. total Software/PM expert time (days)	-	10

3.9.4. Resources Required / Committed

Member Company Resources	*Reqired Days*	*Market Value*		*Days Committed*	*Resource shortfall*
Requirements definition					
Process Model	10	$5 000		10	0
Usage Scenaria	20	$10 000		20	0
Model design					
Object Model development *(w/ tech.Support)*	6	$3 000		?	?
Integration *(w/ tech.Support)*	9	$4 500		?	?
Design and Implementation validation					
Test Case development	5	$2 500		?	?
Review/feedback on implementations	5	$2 500		?	?
Project Management					
Project management and administration	10	$5 000		10	0
Travel and Meetings	10	$10 000		10	0
Total Member Company Resources	**75**	**$42 500**		50 + ?	?

Model/Specification development support	**Required Days**	**Market Value**		**Days Committed**	**Resource shortfall**
Technical support	nn	$nn			
Project management	nn	$nn			
Publication and Administration	nn	$nn			
Equipment and software	nn	$nn			
Travel and subsistence	nn	$nn			

Total Project Support	**nn**	**$nn**			
Total for Project	**nn**	**$nn**			

3.10. [FM-4] Occupancy Planning

3.10.1. Project Description

This project includes the following three processes:

- Occupancy Planning
- Design of Workstations
- Layout of Workstations for an Open Office

The occupancy planner (includes interior designers, facilities managers, architects, furniture dealers' designers, etc.) applies standards during the assignment of people and organizations to interior spaces. It also involves the planning and moving of building assets such as equipment and furniture. This process occurs during the initial planning of space occupancy, and whenever that occupancy needs to change (company reorganization, company growth, or new hires, etc.). The layout and design of typical workstations can be sub-processes of the occupancy planning when it involves systems furniture planning for open offices. These processes require information about the building floor spaces. They also generate space occupancy data for future use of office planning.

Automatic input and utilization of the IFC supported object data, such as building elements and spaces as well as FF&E and occupants, may improve the efficiency of the processes. New objects generated will also be IFC compliant so that they can be used by varies FM processes during the operation of the facility.

3.10.2. Project Team

Project Leader ***Kevin Yu - NA***

Chapter	***Name***	***Email***	***Hrs / Wk***
NA	Rick Bartling / Karen Smith-Hosner	rbartling@hermanmiller.com ksmithhosner@hermanmiller.com	3.5
NA	Vicky Borchers	vicky@mksinfo.qc.ca	7
NA	Francois Grobler	f-grobler@cecer.army.mil	7
NA	Kevin Yu	kevin@naoki.ca	12.5
NA	Rob Wakeling	Robw@visio.com	5
UK	Paul Chadwick	fax: 117-943-4113	?
Total for project			***35***

3.10.3. Scope of Work

AEC processes to be supported	-	3	Est. total AEC expert time (days)	-	29
Expected IFC Model Impact (1 *(min)* to 5)	-	5	Est. total Info Modeling expert time (days)	-	61.5
Degree of technical difficulty (1 *(min)* to 5)	-	4	Est. total Software/PM expert time (days)	-	32

3.10.4. Resources Required / Committed

Member Company Resources	**Reqired Days**	**Market Value**		**Days Committed**	**Resource shortfall**

Requirements definition					
Process Model	15	$4,725		15	0
Usage Scenaria	15	$4,725		15	0
Model design					
Object Model development *(w/ tech.Support)*	30	$9,450		30	0
Integration *(w/ tech.Support)*	15	$4,725		15	0
Design and Implementation validation					
Test Case development	25	$7,875		15	10
Review/feedback on implementations	7.5	$2,363		0	7.5
Project Management					
Project management and administration	15	$4,725		11	4
Travel and Meetings	60	$4,800		60	0
Total Member Company Resources	**132.5**	**$43,388**		**161**	**21.5**

Model/Specification development support	**Required Days**	**Market Value**		**Days Committed**	**Resource shortfall**
Technical support	nn	$nn			
Project management	nn	$nn			
Publication and Administration	nn	$nn			
Equipment and software	nn	$nn			
Travel and subsistence	nn	$nn			
Total Project Support	**nn**	**$nn**			
Total for Project	**nn**	**$nn**			

3.11. [SI-1] Photo Accurate Visualisation

3.11.1. Project Description

In the design of a building or other structure, the architect or designer may want to see what the building or the structure will look like, or may want to render images for the client's benefit. Such visualization may be desired at any time from the earliest architectural design or retrofitting to the final interior design. Visualization is the key to solving lighting and daylighting design problems, and is also important in assessing building performance and human comfort issues. IFC support of this process may reduce input preparation time by 75-85% process (through automatic acquisition of building geometry and all surface properties) and thus make the use of the corresponding applications economically feasible.

3.11.2. Project Team

Project Leader: Vladimir Bazjanac, North American Chapter

Chapter	***Name***	***Email***	***Hrs / Week***
North American	Vladimir Bazjanac	vlado@gundog.lbl.gov	as /possible
U.K.	Sandy Kinghorn	100412.3254@compuserve.com	?

3.11.3. Scope of Work

AEC processes to be supported	-	3	Est. total AEC expert time (days)	-	1
Expected IFC Model Impact (1 *(min)* to 5)	-	1	Est. total Info Modeling expert time (days)	-	1
Degree of technical difficulty (1 *(min)* to 5)	-	1	Est. total Software/PM expert time (days)	-	1

3.11.4. Resources Required / Committed

Member Company Resources	**Reqired Days**	**Market Value**		**Days Committed**	**Resource shortfall**
Process Model	3	$2,250		0	$2,250
Usage Requirements	1	$750		0	$750
Object Model development	.5	$375		0	$375
Integration	0	$0		0	$0
Test Case development	5	$3,750		0	$3,750
Implementation technical support	0	$0		0	$0
Management and Review	1	$750		0	$750
Total Member Company Resources	**10.5**	**$7,875**		0	**$7,875**
Project Support	**Required Days**	**Market Value**			
Technical support	0	$0			
Project management	0	$0			
Publication and Administration	0	$0			
Equipment and software	0	$0			
Travel and subsistence	0	$0			
Total Project Support	**0**	**$0**			
Total for Project	**10.5**	**$7,875**			

3.12. [XM-2] Project Document Management

3.12.1. Project Description

Project Document Management refers to all information pertaining to the documents used to estimate, bid, purchase, and manage the building process as well as for use within the Facilities Management domain. This data identifies the document, the author of the document, changes to the document since the last change, and relationships to other documents.

-

It is being suggested to the group that the first concentration of our work will be on the Contract Drawings represented in the model. It is acknowledged that this is only a small subset of the related documents of the model. We will continue to review the areas affected and complete a framework for our section of work with a complete understanding of what will be reflected in the first pass of our work into the model by the end of our first full meeting to be held at the end of January.

- *Who performs this process?*

All software vendors that use drawings, specifications, and sketches during the life cycle of a project. This would include (the Architect’s use of) CAD, estimating, scheduling, management, and facilities management software vendors.

- *When in the project lifecycle it is performed?*

From the very inception of the project, where these documents are used to define the project, through the construction of the project with all of its changes, through the management of the “building” once the project is complete.

- *What other processes does it relate to (input from/output to/controlled by)?*

This process starts in the creation and modification of the documents and outputs to all processes that use the documents as a means of identification. This would include estimating where changes to the work are usually quantified by document, management, where the documents are used to control the flow of work on a project and establish what is being built by document, and Facilities Management, where documents are the prime method of identifying actual conditions in a facility.

- *What is the benefit (time or cost savings) in IFC based application support of these processes?*

The control of the project over time depends upon the comparison of many baselines of data from one point in time to another. These baselines are reflected as (can be seen as) documents with a reflection in time. Without the identification and use of these documents, such as a Change Estimate, applications would not be able to identify themselves as distinct from others. In this way, applications such as Estimating, Purchasing, Scheduling, and Management packages are enabled to provide these standard views of a project model. In addition, where documents are still being used as the preferred method of delivery of information regarding a project, such as various government agencies requiring drawings and members of the project team who are not CAD enabled.

3.12.2. Project Team

Project Leader Raymond H. Brungard - North American

Please note that the team makeup for this work will be international and cross domain in nature. There are a number of individuals who are interested in this work and I am at this time arranging for the final team size and makeup, without the undue disruption of other groups. It is my intention to make sure that the project team includes members from the CAD and Architectural backgrounds to round out the view of Contract Documents.

Chapter	***Name***	***Email***	***Hrs / Week***
NA	Raymond H. Brungard	rbrungard@tcco.com	7
UK	Graham Storer	G_Storer@tel-consult.co.uk	7
UK	To be named later		4
NA	Ken Herold (part time)	iaiexec	1
	As yet Named CAD Software		7
Nordic	Arto Kiminieri	arto.kiminieri@vtt.fi	7
NA	Mike Cole (part time)		.5
Total for Project			33.5

3.12.3. Scope of Work

AEC processes to be supported	-most	Est. total AEC expert time (days)	- 50
Expected IFC Model Impact (1 *(min)* to 5)	- 2	Est. total Info Modeling expert time (days)	- 5
Degree of technical difficulty (1 *(min)* to 5)	- 4	Est. total Software/PM expert time (days)	- 15

3.12.4. Resources Required / Committed

Member Company Resources	**Required Days**	**Market Value**		**Days Committed**	**Resource shortfall**
Requirements definition					
Process Model	25	$1,250		25	$1,250
Usage Scenaria	25	$1,250		25	$1,250
Model design					
Object Model development *(w/ tech.Support)*	5	$250		5	$250
Integration *(w/ tech.Support)*	5	$250		5	$250
Design and Implementation validation					
Test Case development	10	$500		nn	nn
Review/feedback on implementations	5	$250		nn	nn
Project Management					
Project management and administration	5	$250		nn	nn
Travel and Meetings		$12,000		n/a	nn
Total Member Company Resources	**80**	**$16,000**		nn	nn

Model/Specification development support	**Required Days**	**Market Value**			
Technical support	3	$200			
Project management	5	$340			
Publication and Administration	5	$340			
Equipment and software	2	$130			
Travel and subsistence	5	$340			
Total Project Support	**20**	**$1,350**			
Total for Project		**$nn**			

4. AEC/FM Industry Process Definitions

This section defines the end user domain processes to be supported by this release of the IFC Project Model. Requirements for information to be included in the project model were derived from these end user processes. TQM process diagramming has been used to help document these process definitions.

To further elaborate requirements for software applications that will support these processes, detailed task definitions and user usage scenarios are also provided. In general, these usage scenarios define how the AEC domain professionals expect to be able to use applications (supporting IFC) to accomplish the associated processes specified.

Please note the model validation section which follows. It contains a series of test cases which should enable application developers to test and validate that their applications do indeed satisfy the end user requirements for each process.

As for most sections of this specification, this one is organized by AEC domains in the following order: Architectural design, HVAC engineering, Codes and Standard, Cost Estimating, Facilities Management, Simulation and All Domains.

4.1. [AR-1] Architectural Model Extensions

Processes Defined in this project:

1. Building Shell Design
2. Building Core Design
    3. Stair Design
    4. Restroom Design
5. Roof Design

4.1.1. Process: Building Shell Design

The architect balances the building massing with the elevation aesthetics while performing exterior shell design. Both processes (massing and shell design) evolve and cycle back and forth as each may change aspects of the other. The exterior shell design involves making the massing interesting while using glass fenestration, cladding materials, and details in adornment that create a scale and design motif. Other aspects of this process, that are balanced, are the need for visual access and illumination of the spaces behind the shell, and the issues of attaching and waterproofing the shell. The shell design starts typically after a preliminary space layout and during the building massing studies.

4.1.1.1. Introduction

Overview:

The architect starts the shell design by working with the preliminary stacking and blocking diagrams to determine a massing of the building, based on the floor plates created in the space layout phase. After the massing, the architect will determine the proper aesthetics effect for the building, whether the facade is connected to the outside of the structure or integrated within the structure. The fenestration is determined based on the amount of light and visual impact of the glass and openings on the facade. After the designer determines the type of materials used, preliminary heat gain/heat loss can be calculated for operational cost impact of the building shell. With the final selection of material and fenestration, a detailed design of the adornment of the facade proceeds using reveals, treatment of the materials, cornices, and other building design elements.

Process Scope:

- None defined

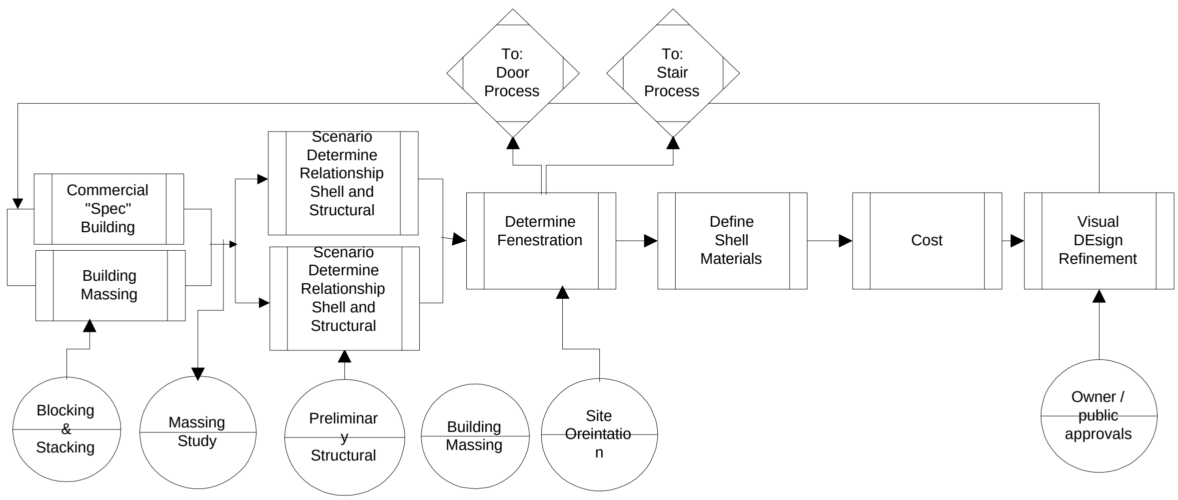
Out-of-Scope:
- block and stacking
- site analysis
- location of the building

Definitions:
- Shell - The exterior wall of a building. Other terms used (facade, elevation, building envelope
- Massing - The exterior shape of a building. A volumetric view of the building

References:
- None defined

Contributors:
- Project team

4.1.1.2. Process Diagram: Building Shell Design



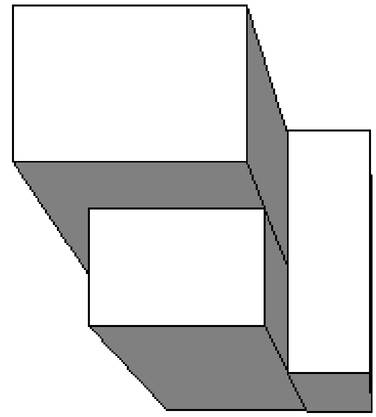
4.1.1.3. Process Definition: Building Shell Design

4.1.1.3.1. Overview

None provided

4.1.1.3.2. Task 1 - Preliminary Building Massing

Task Description:



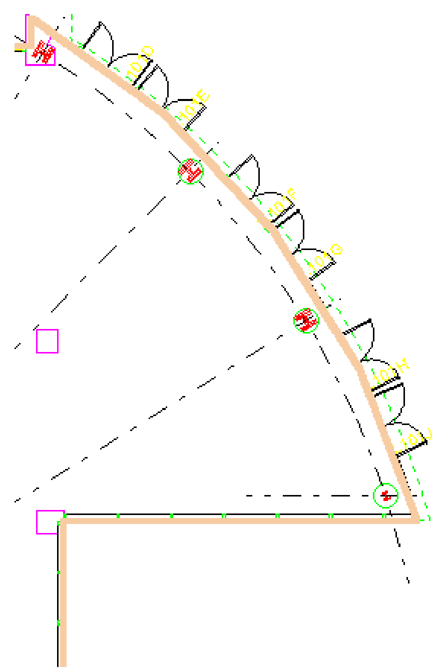
The preliminary building massing is a process that is the definition of the volume of the building shape. The massing may be constrained by regional height restrictions and open area standards which are to balance the open area on a site compared to the building footprint area. The massing will also be driven by considering the size of each floor based on a preliminary block and stacking. Client requirement such as optimizing the amount of the occupational space against the exterior wall or the number of corner offices may suggest a shape to the designer. Other subjective issues such as a desire to step the building down to a human scale may drive the massing and shape of the exterior envelope of the building. The floor to floor height of the interior spaces required by the program has a vertical impact on the massing. At this point in the process the designer will start to think about a preliminary structural grid based on a design.

Example Usage Scenario:

None provided

4.1.1.3.3. Task 2 - Determine the Relationship between Shell and Structure

Task Description:



The relationship of the shell and structure is based on the effect the architect wants to achieve with the design. For example, the shell may be attached to an edge of slab and column so the shell hangs and covers the structure. On the other hand, the designer may desire to express the structure and allow the columns and floor slabs to protrude past the shell, in effect using the structure to frame the shell areas. Other design scenarios such as using the structure to shade glass areas may suggest to the designer to extend the structure past the shell.

Example Usage Scenario:

None provided

4.1.1.3.4. Task 3 - Determine Fenestration

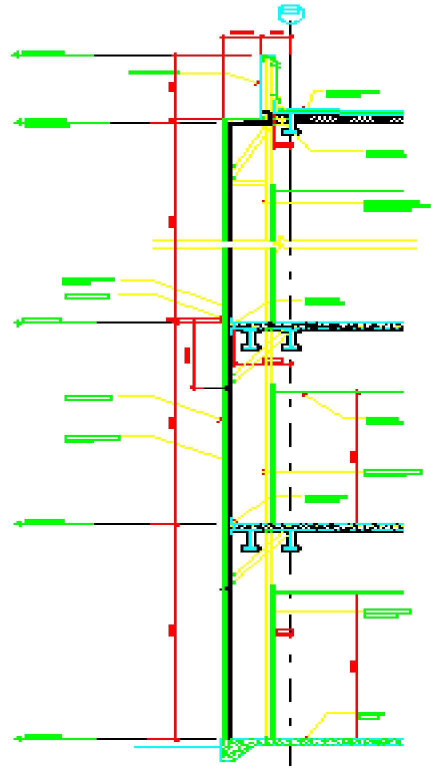
Task Description:

The fenestration is the design and placement of glass area on the shell to permit natural lighting of building spaces and views from the building. The fenestration is based on the rhythm and aesthetics effect the facade should have with respect to glass area. At this stage, a decision on the shape and size of windows are made but not detailed. The amount of glass area may be driven by the energy criteria and regional location and climate. Each facade or elevation of the shell may have a different fenestration due to the orientation of each building face compared to the direction of the sun during different seasons.

Example Usage Scenario:

None provided

4.1.1.3.5. Task 4 - Define Shell Materials



Task Description:

The selection of the shell material is based on a diverse set of criteria. The material may be picked based on the need to fit into other buildings in the area or a regional style or culture. The climate may drive the material selection process along with desires by the client to achieve a style for the building. The durability may create a narrower palate of material. There are also regional construction methods, ease of use, cost, and availability of certain materials that would affect its selection.

Example Usage Scenario:

None provided

4.1.1.3.6. Task 5 - Costs

Task Description:

A preliminary analysis may be run to determine the effect of the shell design on the construction and operational cost of the building. The upkeep on the materials along with the construction cost drive the overall life cycle cost of the shell. On the operational side of the equation the quantity and cost of energy to maintain a temperate environment will be determined by the fenestration and materials selected during the design process. Both will have an overall impact on the heat gain and loss of the building shell.

Example Usage Scenario:

None provided

4.1.1.3.7. Task 6 - Visual Design Refinements

Task Description:

At this point in the process, the shell is refined and detailed. This may include finishes, additions or treatment to materials such as flame/rough/polished stone, reveals, setting back panels, cornices, or parapets. Each of the adornments, construction techniques, and use of materials are used to apply a character to the design of the facade.

Example Usage Scenario:

None provided

4.1.2. Process: Building Core Design

The core design is a balance between making available ancillary spaces and program requirement. The size and location on a floor is determined by the structural systems, program requirements including number of occupants and building codes such as ADA. The design of the core follows the initial layout of the spaces defined in the building program. The spaces that make up the core are typically not defined in the program but are extracted by information about the floor size and occupants.

4.1.2.1. Introduction

Overview:

The core design starts by determining the size of the items needed in the core. Calculations for the number of elevators are based on building occupants and number of floors. The restroom size is based on the number of occupants on the floor and in the building. The floor to floor height is used to determine the length of the stairs which determines the size of the stairwell. The circulation around the core is determined by the type of occupancy and fire codes. The layout of the pieces of the core are driven by the structural grid and distances determined by codes, etc.

Process Scope:

- Assumptions /presumptions: space program (owners' criteria); occupancy, building, floor; parking garage impacts (structural grids); materials handling (site delivery, building services). The core is defined as items for circulation and service delivery for occupants. It does not have to be in the center of the building.

Out-of-Scope:

- This process does not address the actual design of stairs, restrooms, parking design and lobby design. Also materials handling and entering and exiting the building are not included in the core design.

Definitions:

- None defined

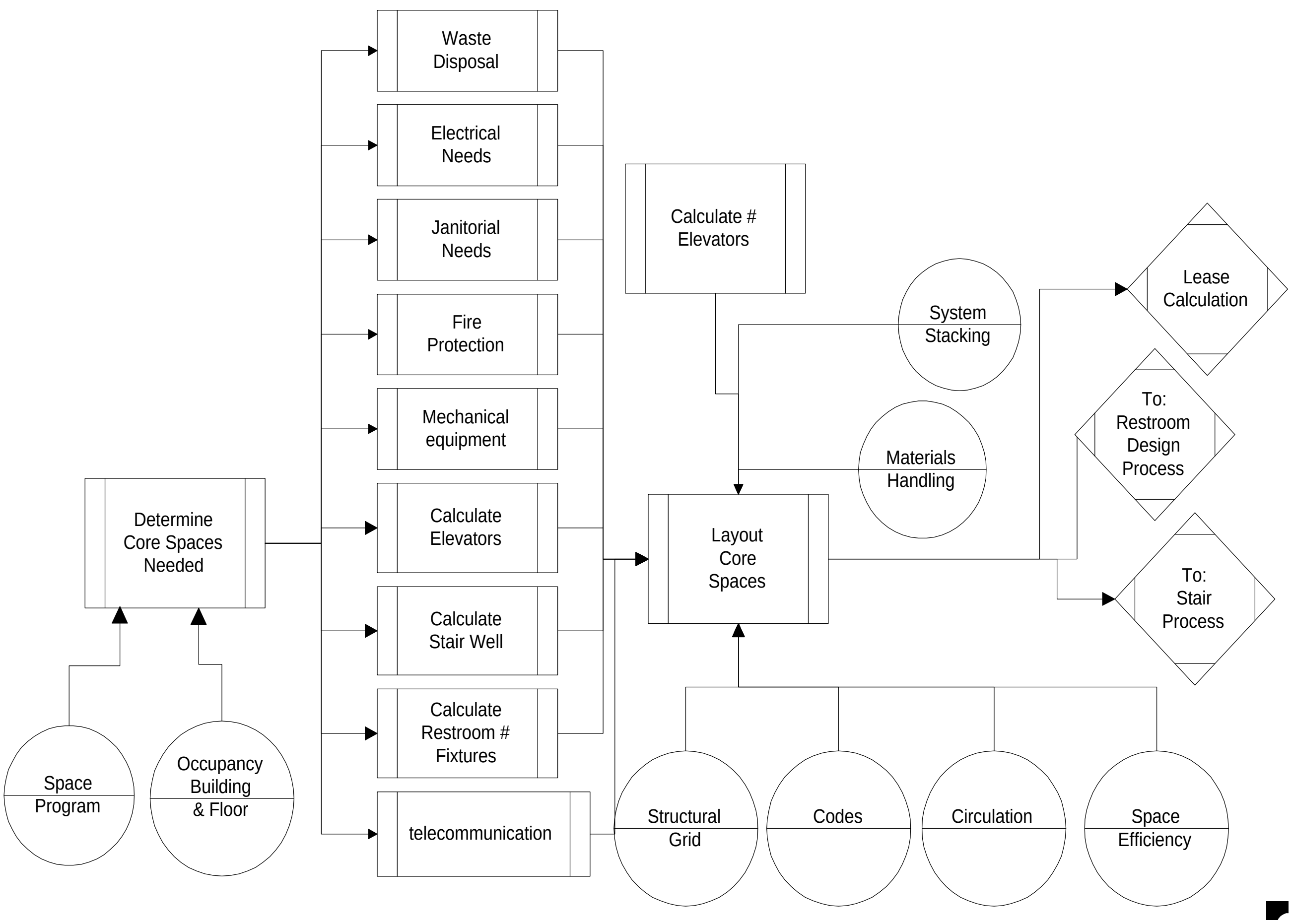
References:

- None defined

Contributors:

- project team

4.1.2.2. Process Diagram: Building Core Design



4.1.2.3. Process Definition: Building Core Design

4.1.2.3.1. Overview

None provided

4.1.2.3.2. Task 1 - Determine Core Spaces Needed

Task Description:

The types of core spaces are determined by a range of issues and codes. The floor occupancy, building type, and building codes determine the type and number of spaces needed as part of the core. The types of building services that are needed in the building will determine additional types of spaces to allow passage and access to services central to the buildings operation.

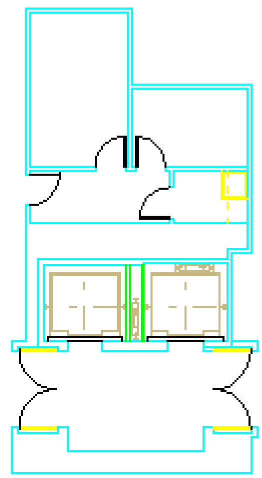
Example Usage Scenario:

None provided

4.1.2.3.3. Task 2 - Determine Core Space Sizes

Task Description:

After the determination of which spaces are included in the core for each floor the overall sizes for each needs to calculated. Apply codes and other processes to determine the size and shape of core spaces. The size of service spaces such as chases and shafts are determined by the overall amount of the material such as fluids, gases, and electrical/Telecommunications that have to be passed through and distributed to floors. Spaces used for transporting occupants such as stairs and elevators are calculated based on the volume of circulation determined by the occupancy of the floor and the building they serve. The final areas provided for occupant support such as restrooms are determined by the occupants of each of the floor they reside on.



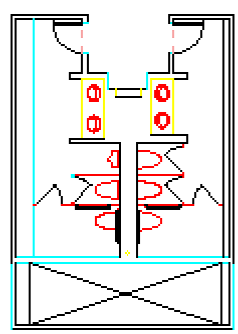
Example Usage Scenario:

None provided

4.1.2.3.4. Task 3 - Layout Core Spaces

Task Description:

The location of the varied spaces in the core is determined by many factors. One of the strongest constraints is the circulation needs for both providing effective space utilization and egress/access to the floor through stairs and elevators. The loads and timing of occupant circulation will determine the number of cabs and ultimately the number of elevator stacks and size of their corresponding shafts. The need to efficiency stacking building services forces the stacking of spaces. The structural needs for sheer walls and the spacing of vertical elements such as columns affects the placement of spaces. If the building includes levels of parking, the trade off between structural bay size and efficient parking layout to optimize the number of parking spaces will affect core element placement.



Example Usage Scenario:

None provided

4.1.2.3.5. Task 4 - Detailed Design of Stairs

Task Description:

Covered in this document under Stair design Process.

4.1.2.3.6. Task 5 - Detailed Design of Restrooms

Task Description:

Covered in this document under Stair design Process.

4.1.3. Process: Stair Design

Stair design is accomplished by working with the major elements, such as treads, landings, and railings, to determine the appropriate size of the stair and its elements. The process is an iterative process where the answer for one of the elements may change the size of another. The two factors that determine many of the size related decisions are based on the occupancy load and the exiting requirement.

4.1.3.1. Introduction

Overview:

The architect starts the stair design by working with information about the building such a location of the stair based on egress. The width and depth is defined during a process of working back and forth. The width is determined by the number of occupants traveling through the stairwell during an emergency. The width is typically defined in the local building and fire codes. The floor to floor heights of the story are used to determine the length of the stairs, based on a rise and run. The designer may then design the depth of the landing based on codes. As the design progresses to the handrail, its design can potentially affect the width of the stairs and landing, depending on the distance it protrudes into the stairwell. At the point where the size of the treads, landing, and the handrail are set, the materials and construction methods are determined. The final design involves adding items such as exit signs, doors and hardware, and emergency lighting.

Process Scope:

- The process described is for fire stairs in a building. Include fire stair materials. ADA safe haven concept should be included (telecommunications, extra design space, area impact)

Out-of-Scope:

- Ornamental stairs not in scope and not required for exiting a floor, ladders.

Definitions:

- ADA safe haven

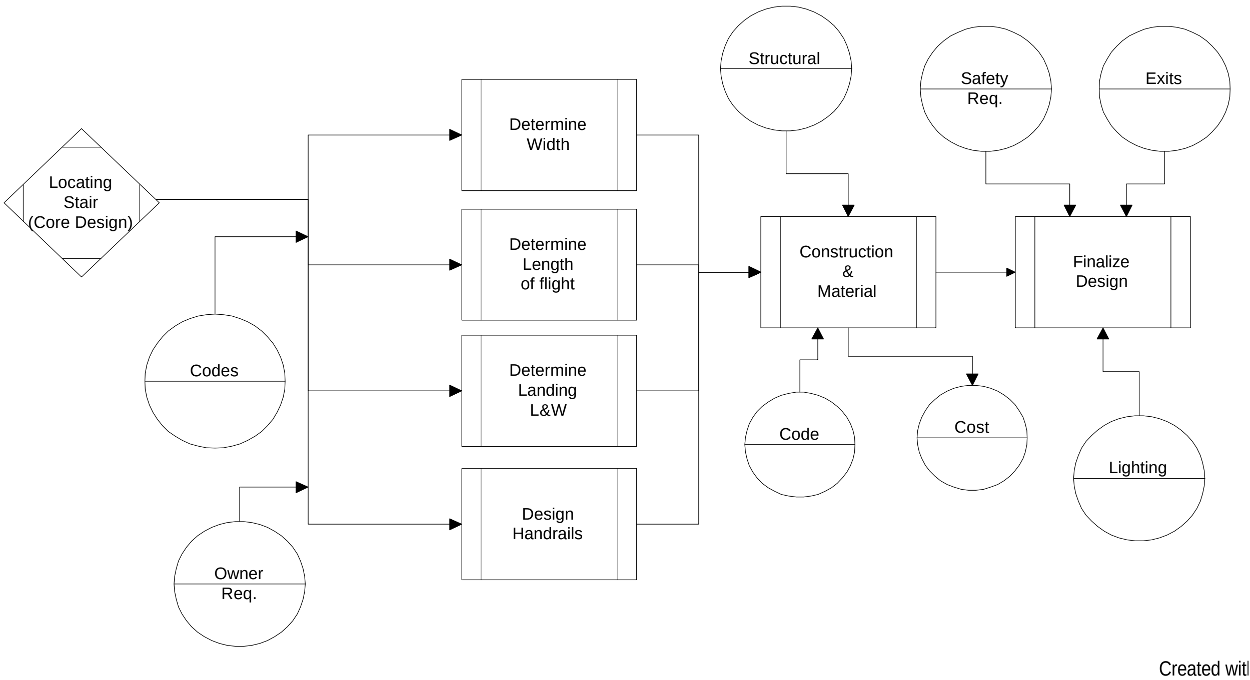
References:

- Safe haven documentation
- Calculation of stair rise and run

Contributors:

- project team

4.1.3.2. Process Diagram: Stair Design



4.1.3.3. Process Definition: Stair Design

4.1.3.3.1. Overview

The blocking and stacking process as an element of conceptual design begins after a building program is defined between the client and architect. The designer starts by creating graphic spaces according to the sizes defined in the building program. While reviewing the adjacency and space size, the spaces are moved around to determine their location horizontally on a floor in the building. The non-programmed spaces such as grouped core elements and circulation are added to the diagram. The process progresses when the vertical location of the space in the building (i.e. stacking is determined). The architect moves between the blocking and stacking tasks until the spaces are organized in an optimal manner. The building structural grid may be refined during this iterative stacking and blocking process.

4.1.3.3.2. Task 1 - Locate Stairs

Task Description:

Covered in this document under the Core design Process

4.1.3.3.3. Task 2 - Determine Width

Task Description:

The width of the stairs are determined by building codes which indicate the minimum sizes based on the number of occupants using the stairwell over a certain amount of time. The designer should take into consideration the depth of the handrail as it protrudes into the stair and cuts down on the actual width of the tread.

Example Usage Scenario:

None provided

4.1.3.3.4. Task 3 - Determine Tread and Risers

Task Description:

The length or run of the stairs is dependent on the height between the floors being calculated. There are appropriate height and depth of treads based on what is comfortable for occupants to walk up and down steps without stumbling. The rise/run of the stairs are defined in tables in local building codes.

Example Usage Scenario:

None provided

4.1.3.3.5. Task 4 - Determine Landing

Task Description:

The landing performs two functions. First it allows the occupants a place to exit out of a floor onto the stair well. The second function is that it is a location to change directions in the stair well. The landing width and depth is determined by stairs connected to the landing and the number of occupants switching between stair flights. The local building codes describe the appropriate size based on the occupants on each of the floors. A new requirement is the inclusion of a safe haven, which is an alcove on the stair landing where a wheel chair can reside out of the way of stair traffic until help can arrive.

Example Usage Scenario:

None provided

4.1.3.3.6. Task 5 - Guardrail design

Task Description:

None provided

Example Usage Scenario:

None provided

4.1.3.3.7. Task 6 - Handrail design

Task Description:

None provided

Example Usage Scenario:

None provided

4.1.3.3.8. Task 7 - Construction and Materials

Task Description:

As the design of the stair is taking shape, a decision on materials is made. The designer selects the material for the stairs such as concrete, steel, or a combination of both. The decision may be based on regional standards, ease of construction, or local fire codes. The materials on the tread and the type and construction of the nosing are also made at this point in the process. The final stage of deciding on the construction the designer determines how the stringer connects the tread, riser, and connects it to the stair well.

Example Usage Scenario:

None provided

4.1.3.3.9. Task 8 - Finalize Design

Task Description:

The final detail of stair design evolves other objects connected or part of the stair. This may include deciding on the type of exit doors, signage, standpipe location, location of vents and hatches. Also design of emergency lighting and ventilation should be performed by fire safety engineers at this point in the process.

Example Usage Scenario:

None provided

4.1.4. Process: Restroom Design

The design of restrooms involves effective movement of building occupants, ADA codes, and aesthetic use of materials. The minimum number of fixtures is determined by the number of occupants that reside on a floor or visit a floor.

4.1.4.1. Introduction

Overview:

At the start of restroom design, the number of fixtures are determined by the floor occupancy. The designer will also determine items such as partition type, fixture type, stall sizes, based on codes such as ADA and any client requirements. The next level of design involves locating the restroom fixtures and lavatories to use the most effective amount of space to contain cost but provide effective circulation. The next level of design involves locating the lavatories, mirrors, towel racks, grab bars, hand dryers, and any other object that services the restroom occupants. Appropriate location of fixtures and other items in the restroom may be determined by effective use of other building services such as plumbing stacks, etc. The final step of design is more aesthetic in that it involves the visual character of the restroom in selecting material type, sizes and objects such as faucets etc.

Process Scope:

- Commercial Public Restroom associated with the building core

Out-of-Scope:

- Locker Rooms, Showers.

Definitions:

- None defined

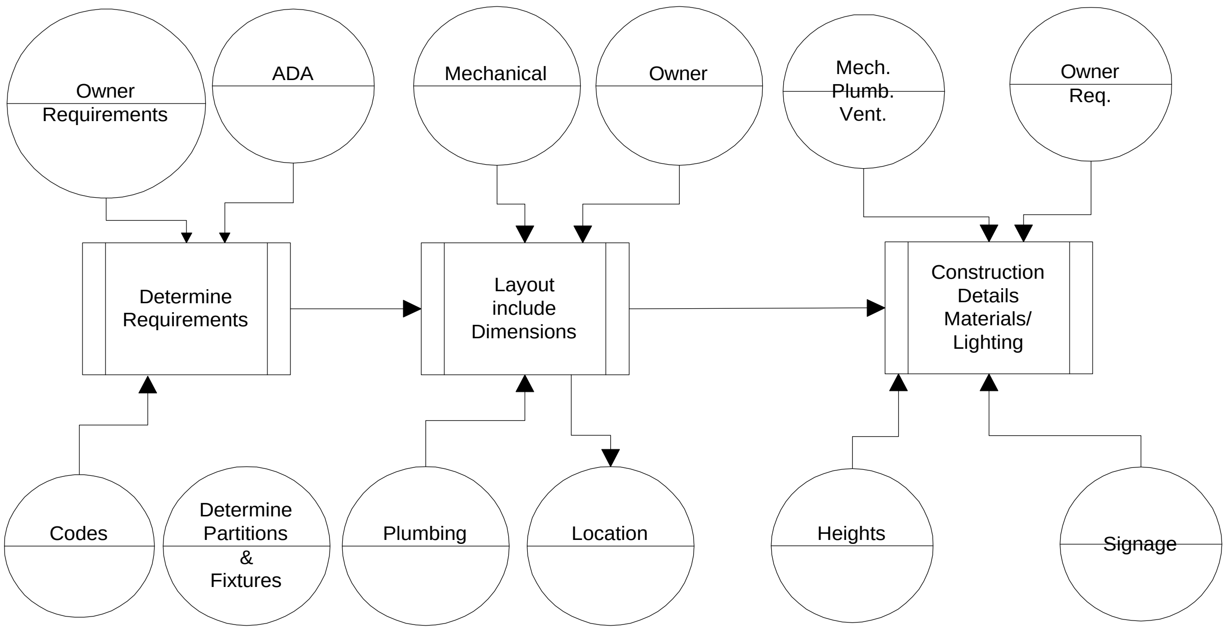
References:

- None defined

Contributors:

- project team

4.1.4.2. Process Diagram: Restroom Design



4.1.4.3. Process Definition: Restroom Design

4.1.4.3.1. Overview

None provided

4.1.4.3.2. Task 1 - Determine Requirements

Task Description:

The number of fixtures which is considered toilets, urinals and sinks is determined by codes and the floor occupancy. The ADA requirements define how many of the fixtures are designed for handicapped access.

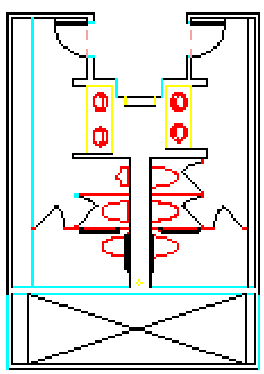
Example Usage Scenario:

None provided

4.1.4.3.3. Task 2 - Layout

Task Description:

Layout involves the location of the major fixtures and the stalls that surround them while creating appropriate circulation for occupants and handicapped. The effective delivery of services such as water and getting rid of waste will set a common plumbing wall which makes is cost effective by stacking all plumbing services for the building.



Example Usage Scenario:

None provided

4.1.4.3.4. Task 3 - Construction Detailing, Finishes and Lighting

Task Description:

hand dryers, trash receptacles, outlets, etc. A closer look at other trades, such as Plumbing, HVAC, and Electrical. The final step of the restroom design involves selecting the materials and lighting appropriate for the building type and clients' requirements. The selection of the style of partitions, faucets, and other fixtures such as whether the toilet is wall hanging or rests on the floor is based on the designer's preferences.

Example Usage Scenario:

None provided

4.1.5. Process: Roof Design

The process of roof design is a mixture of aesthetics, weather dissipation, and hiding other building objects such as telecommunications, mechanical, and elevators. The process is iterative, the designer works back and forth between the massing and roof design to create a building design which expresses a character appropriate to the area, client wishes, and building type.

4.1.5.1. Introduction

Overview:

The architect determines a type of roof based on the design direction and the character of the building. Using the building massing, the architect lays out the roof. On pitched roofs, refinement of the intersection of the roof planes will be necessary. The architect then determines and designs the drainage. The intersection of the roof with the elevations are designed and detailed. The layout and penetration of other services that are hosted on the roof are considered. Materials are selected.

Process Scope:

- Design inputs would cover the process of exterior and interior programs including eaves and overhangs. Interior issues need to address cathedral ceilings, dormers, etc. Exterior roof issues include steeples, parapet roof ventilation, electrical, drainage, recreational areas, planters, irrigation, window washing, skylights, smoke evacuators, access hatches, mechanical screens, roof walk pads, lighting control, and FAA lighting.

Out-of-Scope:

- Actual design of electrical, venting, access hatches, smoke evacuators, sidewalk protection canopies.

Definitions:

- Dormers (space projection from sloped roof, may be considered standard roof, not unique)
- Recreation areas
- Helipads
- Steeples can also be used as a screen or just ornate
- Screening
- Chimneys
- Vents
- Drainage
- Telecommunications: Transmission Tour

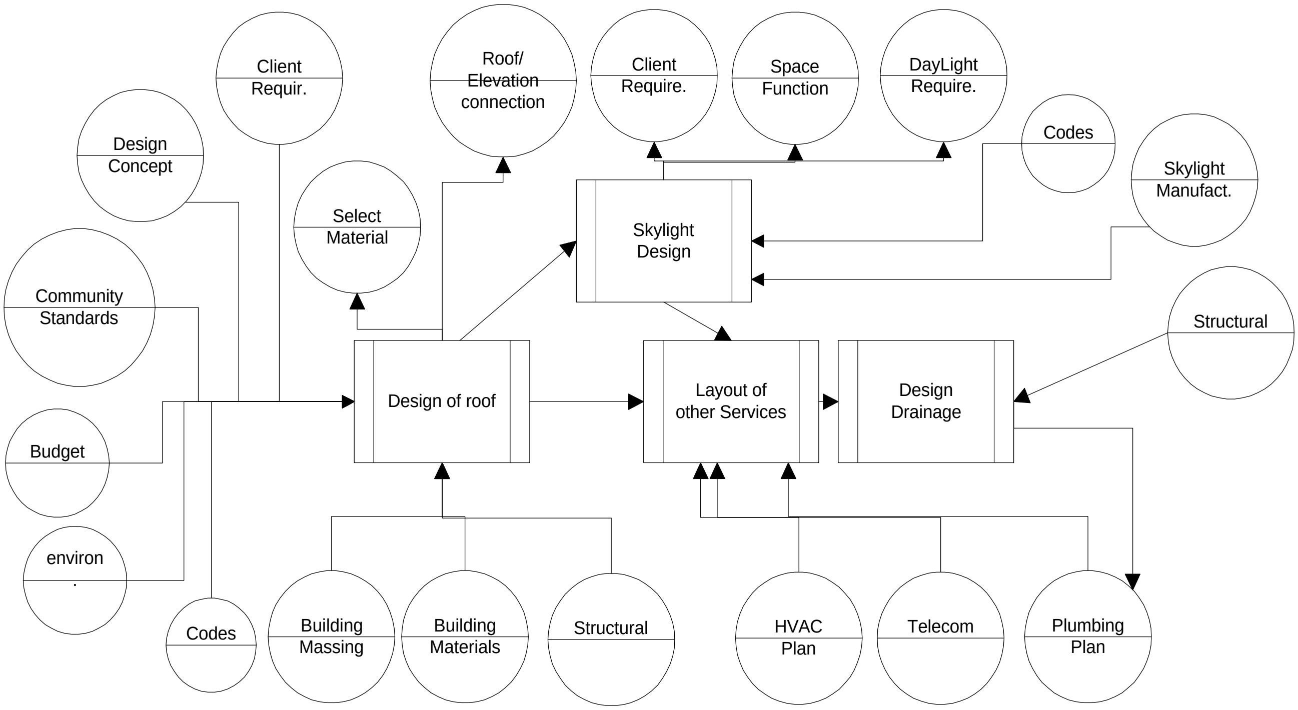
References:

- None defined

Contributors:

- project team

4.1.5.2. Process Diagram: Roof Design

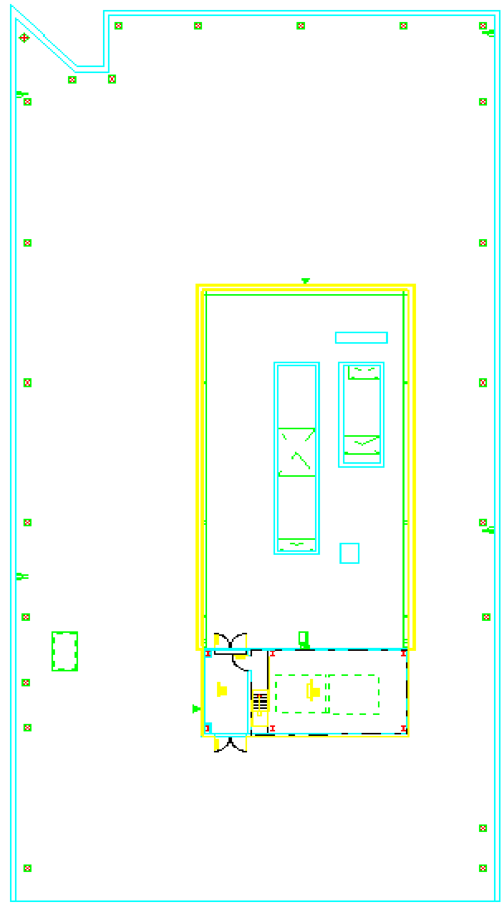


4.1.5.3. Process Definition: Roof Design

4.1.5.3.1. Overview

None provided

4.1.5.3.2. Task 1 - Design Roof



Task Description:

The determination of roof type is a balance between form and function of the building. The decision on style or shape of the roof is a combination of styles of the neighbor buildings along with the desire of the client. The roof type refers to flat, pitched, gabled, etc. An understanding of the types of services supported by the roof may determine the type of roof selected. The regional climate may dictate a shape of the roof structure to support the amount of wind, precipitation, snow, and also radiation of heat from the sun. After the selection of the roof type, a preliminary design is produced to determine the actual shape and its impact on the building form. The slopes of the roof elements to provide the correct shedding of the climatic element will determine pitches. The changes in the massing elements will force the roof to change as new building masses intersect each other. The function of the spaces below the roof may determine the shape along with the need to enclose building services. The type of material used will have a direct impact into the shape of the roof depending on the material constructability. Finally the surrounding building roof-scapes may dictate a direction for the shape.

Example Usage Scenario:

None provided

4.1.5.3.3. Task 2 - Skylight/Clear Story

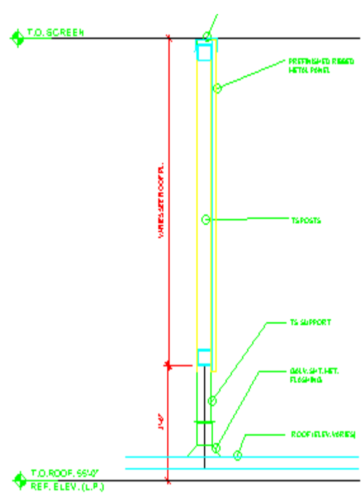
Task Description:

After the shape is created, the integration of any skylights or clear story windows will be integrated into the roof to evaluate the impact and location based on preliminary structural needs. A skylight may not be as simple as a pre-manufactured domed square skylight but could be a complicated barrel vault that runs the length of the building. The intersection of the skylight with the roof becomes critical and may force certain decisions on pitches of roof plains to direct the outside elements away from the glass area.

Example Usage Scenario:

None provided

4.1.5.3.4. Task 3 - Layout of Services



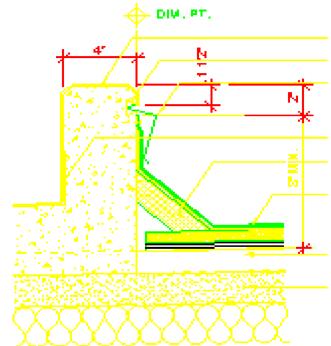
Task Description:

With the major roof shape determined and items such as skylights, etc. placed, the designer then looks at the projections through the roof of items such as vents, stair/elevator, telecommunications, glass cleaning, and mechanical. Depending on the size of the projection techniques such as providing screens and other methods to hide the services may be required. Depending on the building program areas such as heliports, health and fitness, and walkways my be required to be included in the roof design.

Example Usage Scenario:

None provided

4.1.5.3.5. Task 4 - Design Rain/Snow Drainage



Task Description:

At this point, after the building services are located the shedding of water needs to be addressed. The runoff of water is calculated based on the roof planes and slopes and a design concept is created to use roof drains, scuppers, or gutters to empty the water from the roof. The weather (water, snow, heat) are the major cause of roof failure. Details are created to communicate how to keep moisture out of the building and delineate the intersection of materials at joints.

Example Usage Scenario:

None provided

4.2. [AR-2] Compartmentation of Buildings

Processes Defined in this project:

1. Compartmentation of buildings

4.2.1. Process: Compartmentation of buildings

4.2.1.1. Introduction

Overview:

The overall process is split into two stages as defined below:

Stage 1 is concerned with limiting the spread of fire and smoke within the building.

- The building is sub-divided into compartments, enclosed by fire resisting construction.

The building can be sub-divided into compartments by any or all of the following constraints:

- Main Uses
- Spaces occupied by individual owners and/or tenants
- Regulatory geometrical limits set on floor area or space volume.

Stage 2 is concerned with providing a satisfactory means of escape from a building or part of a building to a place of safety.

- By providing enough Escape Routes of the correct capacity, and which are adequately lit and suitably located.

  A satisfactory Means of Escape is provided by ensuring :

  - *There is an adequate number of exits, to serve a known number of occupants, within a Space, Compartment or Storey.*
  - *There are enough Escape Routes of adequate capacity within a Space, Compartment or Storey, to serve all the occupants, likely to use the route in the event of fire.*
  - *That, if any Space, Compartment or Storey is likely to render any Escape Route unusable due to the emission of smoke, then either, smoke seals to exit doors, or a pressurized lobby to control the passage of smoke are provided.*
  - *That Escape Routes are adequately spaced to limit the travel distance to the nearest exit.*
  - *That Escape Routes are adequateley lit by means of an independent power supply.*

This IFC R2 project will focus on Stage 1 -- the process for identifying fire compartments and the fire protection at their notional boundaries, in order to limit the spread of fire and smoke within the building.

Process Scope:

- None provided

Out-of-Scope:

- Fire Protection to Elements of structure.
- Fire Protection to Electrical, Mechanical and Plumbing Services.
- Fire Fighting Equipment
- Fire Resistance and Surface Spread of Flame
- Interrelationship with adjoining buildings and to the boundary.

Definitions:

- **Fire Use Classification:** classification listing the different possible uses of a building or space for the purposes of fire compartmentation
- **Fire Use:** A member of a Fire Use Classification.
- **Main Use Space:** Is a building or part of a building considered significant for Fire Compartmentation.
- **Ancillary Use Space:** Is a part of a building which supports a Main Use and which is not considered significant for Fire Compartmentation. An Ancillary Use may be considered a Main Use in its own right if it meets certain criteria.
- **Fire Compartment:** A building, or part, comprising one or more spaces constructed so as to prevent the spread of fire to or from another part of the same or adjoining buildings and which meets the area, volume, or occupancy limits set by Statutory Requirement.
- **Single Occupancy Space:** A space possessed and used by only one person or organisation.
- **Height Above Grade:** Height of floor of top storey of Fire Compartment above accessible horizontal surface external to the Fire Compartment.
- **IsSprinklered:** The building is filled throughout with an automatic sprinkler system.

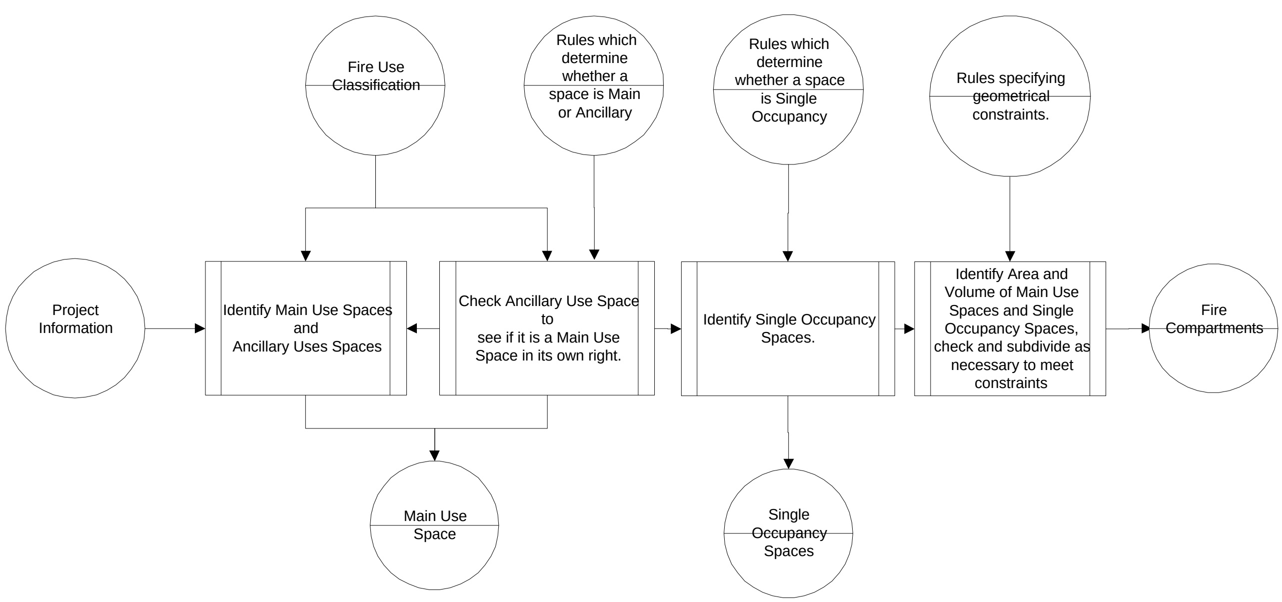
References:

- Building Regulations 1991. Approved Document B: Fire Safety.
- Uniclass classification.
- Ci/SfB classification - space classification.
- British Standards.

Contributors:

- Project Team (see project summary)

4.2.1.2. Process Diagram: Compartmentation of buildings



4.2.1.3. Process Definition: Compartmentation of buildings

4.2.1.3.1. Overview

It is assumed that the fire usually starts in one place, and spreads to other parts of the building. In order to allow the occupants of the building to escape, the first priority is to stop the spread of fire and smoke to other parts of the building, as well as to maintain common escape routes free of fire and smoke. Compartmentation controls fire within a limited space, allowing occupants of the building to escape from the seat of the fire to a place of safety.

Note: Tasks A through D are repeated for each building storey.

4.2.1.3.2. Task A - Identify Main/Ancillary Use Spaces

Task Description:

For each building identify the Main Use Spaces and Ancillary Use Spaces.

Example Usage Scenario:

None provided

4.2.1.3.3. Task B - Adjust Main/Ancillary Use Spaces according to Code

Task Description:

Analyse the Ancillary Use Spaces with reference to regulations to see if they need to be treated as Main Use Spaces in their own right. If not subsume Ancillary Use Spaces into their corresponding Main Use Spaces.

Example Usage Scenario:

None provided

4.2.1.3.4. Task C - Identify Single Occupancy Spaces

Task Description:

For each building identify all the Single Occupancy Spaces contained within the building.

Example Usage Scenario:

None provided

4.2.1.3.5. Task D - Check Areas/Volumes to Design Fire Compartments

Task Description:

Specify the boundaries of Main Use Spaces and Single Occupancy Spaces as Fire Compartment boundaries.

Analyse the Fire Compartments defined by the boundaries specified in task C. For each compartment check the regulations governing the maximum dimensions for the Fire Use into which its use is classified. Subdivide each Fire Compartment as necessary in order to meet the rules. NB. the maximum allowable dimensions of a Fire Compartment are likely to also depend on whether or not the building in which the compartment is located is sprinklered and on the height above grade of the top floor of the building in which the Fire Compartment is located.

Example Usage Scenario:

None provided

4.3. [BS-1] HVAC System Design

Processes Defined in this project:

1. HVAC Duct System Design
2. HVAC Piping System Design

4.3.1. Process: HVAC Duct System Design

4.3.1.1. Introduction

Overview:

See Project summary.

Process Scope:

- Select and locate components to be connected in the duct system
- Connect components with ducts and fittings
- Locate other system components: dampers, etc.
- Facilitate sizing ducts and fittings
- Facilitate interference checking
- Facilitate pressure loss calculations
- Facilitate fan selection
- Generate final system representation

Out-of-Scope:

- Selection of system type and system configuration
- Actual sizing the duct and fittings
- Performing interference checks
- Performing pressure loss calculations
- Performing fan selection

Definitions:

- ASHRAE - American Society of Heating Refrigeration and Air Conditioning Engineers
- SMACNA - Sheet Metal and Air Conditioning Contractor’s National Association
- BACnet – Building Automation and Control Network

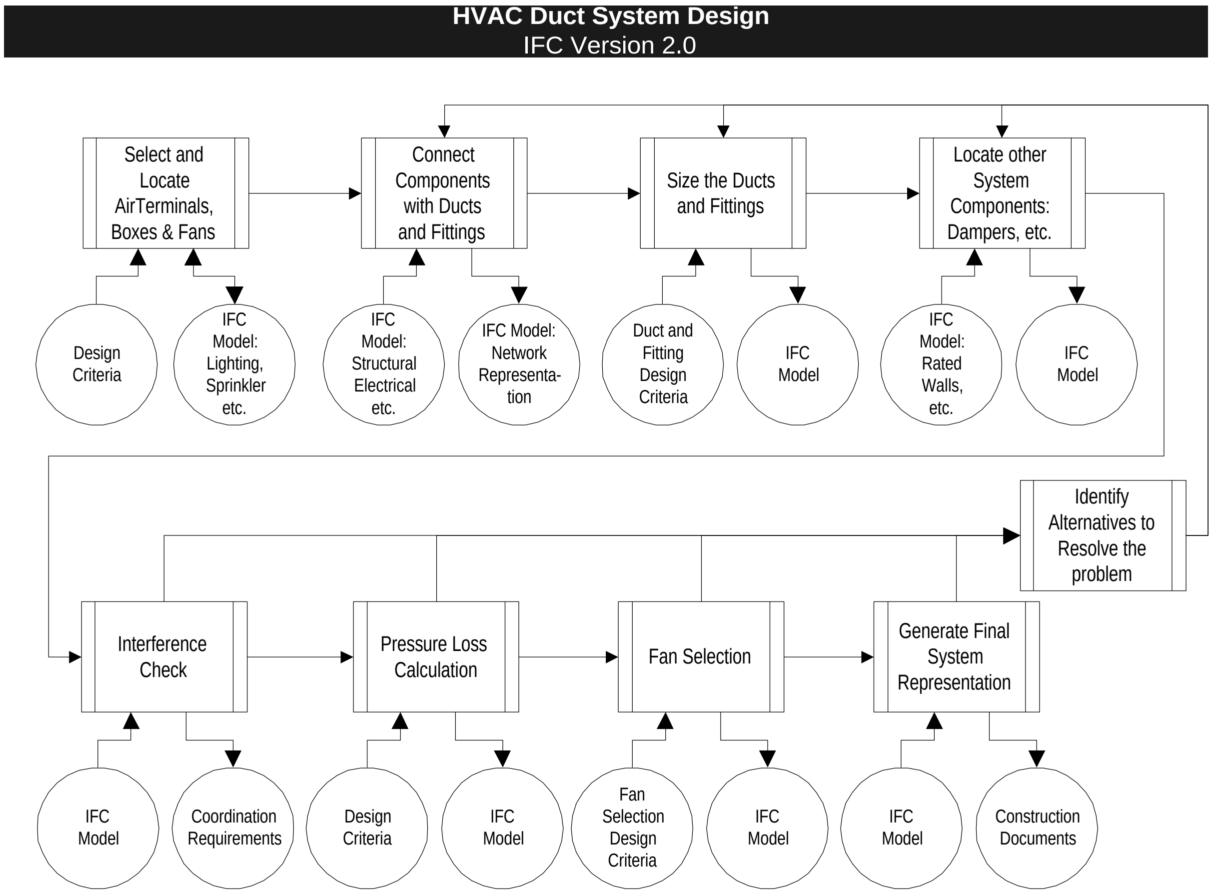
References:

- ASHRAE Handbooks
- SMACNA HVAC Duct Construction Standards
- BACnet Specifications (ANSI/ASHRAE 135-1995)
- IFC R2.0 BS-3 Project: Pathway Design and Coordination

Contributors:

- James Forester, North American Chapter: Technical Coordinator
- North American Chapter -- Building Services Committee
- United Kingdom Chapter -- Building Services Committee
- German Chapter -- HVAC Committee
- Nordic Chapter -- HVAC Committee
- French Chapter -- HVAC Committee

4.3.1.2. Process Diagram: HVAC Duct System Design



4.3.1.3. Process Definition: HVAC Duct System Design

4.3.1.3.1. Overview

Once an appropriate system type has been determined (outside of scope), the HVAC Duct System Design process begins by selecting and locating air terminal devices, air terminal boxes and fans that will be part of the system. Reflected ceiling plans may be available showing light fixtures, sprinklers and the ceiling grid to aid in the location of air terminal devices. If these are not available the engineer selects locations for the air terminal devices and submits the locations to other members of the design team for coordination. To appropriately locate the air terminal boxes and devices, structural information is required so that initial interferences may be avoided.

The next step is to connect the air terminals, terminal boxes and fans together with ducts and fittings. A network representation of this system layout is typically generated for use in calculating duct sizes and a graphical representation is generated for coordination with other disciplines.

The room air flow rates are then assigned to the air terminals. These air flow rates are determined by the building cooling and heating load calculations; these processes are defined in the IFC 1.0 Specifications.

The duct and fitting sizes are then calculated based upon these air flow rates and the duct system design criteria. The duct and fitting sizes are then updated in the graphical representation of the system.

Other required system components (i.e., dampers, sensors, etc.) are then located on the graphical representation. This process requires various architectural information such as the locations of fire rated walls, exit corridors, etc., which are available from the architectural plans. Any components that require other disciplines to respond are identified, such as electrical power required to fan powered terminal boxes.

Once these components are located, an interference check is performed. This requires the coordination with the other building disciplines and may require resizing or relocating ducts, fittings, etc.

A final duct system pressure loss calculation may be required beyond that made during the duct sizing based on changes from estimated values to actual values that can only be determined after the duct sizes are finalized. With the final pressure loss, the total air flow and the engineering design criteria, a fan can be selected.

Primary difficulties in the duct system design process are coordination with other disciplines to prevent conflicts for space and to predict sound levels that result from air flow in the ducts and air terminals.

4.3.1.3.2. Task A - Select and Locate System Components

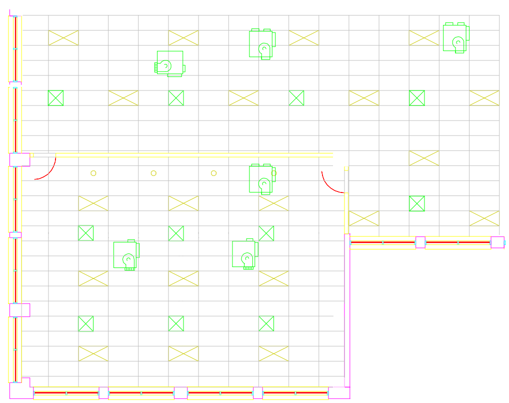
Task Description:

Before the engineer can select and locate system components, the following information is required from other members of the design team:

- Ceiling grid plans (architect)
- Reflected ceiling plans (architect)
- Lighting plans (architect or lighting designer)
- Structural plans (structural engineer)
- Plumbing plans (plumbing engineer)
- Piping plans (HVAC engineer)
- Sprinkler plans (fire protection engineer)
- Smoke detector plans (fire protection engineer)
- Speaker plans (architect or electrical engineer)

With the above information, the engineer can perform the tasks required for this process. If some of the above information is unavailable, the engineer must either generate it manually or make assumptions based on the function and usage of the spaces involved in the design.

Generally there are three types of system components to be selected and located for this process:



- **Fan(s):** The location of the fan(s) used for moving the air in the duct system. The fan(s) may be for supplying, returning or exhausting air from the building or space. The fan(s) may be stand alone or part of a manufactured assembly, which may include coils, filters, mixing boxes, etc. Combination fans, coils etc. may be factory assembled or assembled at the project site. The exact size and capacity of the fan(s) are not required at this stage, though an approximate fan size is necessary to ensure the space selected for the fan is adequate. Though not essential, having the size of the fan outlet is useful in sizing the transition between the fan outlet and the duct.
- **Air Terminal Boxes:** Depending on the type of HVAC system, the system may or may not have air terminal boxes. Terminal boxes are typically located in a branch duct downstream from the main supply duct. There are several different types of terminal boxes. They are used in various ways to control the amount and or temperature of the air being supplied to one or more spaces with similar heating and cooling load characteristics. It is desirable but not necessary to know the exact terminal boxes being used in order to size the associated ducts. If the exact terminal box being used is known, the exact duct connection size and pressure loss through the terminal boxes are known. Also terminal boxes from different manufactures have different dimensions and knowing the exact dimensions and clearances required for maintenance can prevent future conflicts for space.

  Terminal boxes are typically located after the air terminal devices used for distributing the air in the spaces are located. This allows the terminal boxes to be positioned to permit the shortest duct runs between the terminal box and the air terminal devices it supplies.

- **Air Terminal Devices:** Air terminal devices are used to distribute the air from the duct system to the spaces or to remove air from the spaces. The air terminal device can be connected to the supply, return or exhaust air ducts in different ways:

  Directly into the side of a main or branch duct or on a short duct section that allows for a volume damper and/or a lower resistant transition from the duct to the air terminal device. This type of connection is used where the duct is exposed in the space.

  Directly on the outlet of a terminal box.

  On the end of a branch duct from the main duct or from a duct on a terminal box. The air terminal device may terminate in an opening in a ceiling or wall, or be exposed entirely in the space.

  An air terminal device can simply be located within an opening through a wall that forms a chase that is part of the general building construction or to an above ceiling plenum used for transferring return air. Locating an air terminal devices used in this way are not required for sizing the duct system, although they are usually located at the same time other air terminal devices are located.

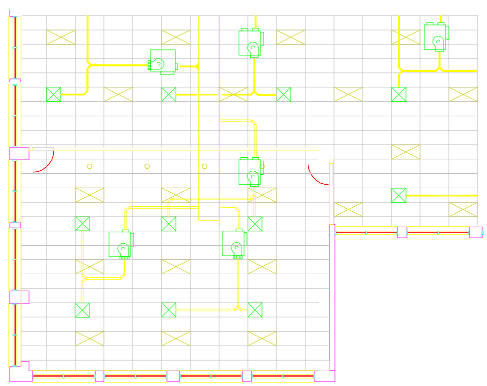
  Selecting the exact air terminal devices and their accessories at this stage is not required to size the duct system but is desirable to keep from revisiting each of the air terminal devices a second time. Making a selection also supplies the exact duct connection required and the exact pressure loss through the air terminal device which is necessary in the final design of the duct system for the fan selection.

Example Usage Scenario:

None provided

4.3.1.3.3. Task B - Connect Components with Ducts and Fittings

Task Description:



Before the engineer can connect air terminals, boxes and fans with ducts and fittings, the following information is required from other members of the design team:

- Floor plans (architect)
- Structural plans (structural engineer)
- Coordination requirements from any other disciplines

This step involves preparing drawings that schematically represent the system under design. The duct is typically drawn from the fan to the air terminal boxes, if any, and then to the air terminal devices. Various types of elbows, tees and other fittings are utilized so each fitting must be designated as to what type is being used. These schematics are then used to begin coordination with other disciplines which are impacted by the duct system. The information derived from the air flow associated with each air terminal device together with the schematic drawing is used by a duct sizing program to determine actual duct sizes. Often the duct connections to different types of equipment are standardized.

Example Usage Scenario:

None provided

4.3.1.3.4. Task C - Sizing the Duct and Fittings

Task Description:

The ducts are sized using the information derived from the schematic drawings and the design criteria established by the engineer. Design criteria include such information as type of design (constant pressure, static regain, etc.), maximum velocity, maximum height of duct, material to be used, etc. The actual methodologies and algorithms used for sizing of the duct and fittings is out of scope, as this is application specific.

Example Usage Scenario:

None provided

4.3.1.3.5. Task D - Locate Other System Components

Task Description:

The location of other duct system components is determined from information in the schematic drawings and the design criteria established by the engineer. Other components, such as fire dampers, volume control dampers, louvers, filters, etc., are then located on the drawing. These components have pressure losses that may only be precisely determined after the actual duct sizes are known. After these pressure losses are determined, the total pressure loss is calculated. In many cases the pressure loss for these components are known before the ducts are sized or can be closely estimated so they can be entered before sizing the ducts and fittings.

Control elements, such as sensors, actuators and controllers, can also be specified at this point. The design engineer typically defines the general parameters of a control device. However, a control system vendor may utilize many different mechanisms to achieve the desired effect intended by the design engineer. For this reason, only a subset of control element information is necessary.

Example Usage Scenario:

None provided

4.3.1.3.6. Task E - Interference Check

Task Description:

Before the engineer can perform an interference check, the following information is required from other members of the design team:

- Floor plans (architect)
- Ceiling grid plans (architect)
- Reflected ceiling and/or lighting plans (architect or lighting designer)
- Power plans (electrical engineer)
- Piping plans (HVAC engineer)
- Plumbing and sprinkler plans (plumbing or fire protection engineer)
- Structural plans (structural engineer)
- Coordination requirements from any other disciplines

Interference checking identifies where changes are required in the location or size of specific ducts in order to eliminate physical conflicts with other building components or systems. An example scenario could be that the height of a duct is too great, thus requiring a transition fitting to clear a beam or a pipe. After any interferences are corrected, the total pressure loss for the system can be calculated. Performing the actual interference check is out of scope, as this is application specific.

Refer also to the IFC R2.0 BS-3 Project: Pathway Design and Coordination.

Example Usage Scenario:

None provided

4.3.1.3.7. Task F - Identify alternatives to design problems

Task Description:

This step requires the designer to go back and redesign certain portions of the system. This may involve regenerating the schematic design documents and recalculating system component sizes. Note that this step may occur at any point in the process.

Example Usage Scenario:

None provided

4.3.1.3.8. Task G - Pressure Loss Calculations

Task Description:

After any interference conflicts are corrected the total pressure loss for the system can be calculated. This information is essential to properly select a fan that will serve the duct system. Performing the actual pressure loss calculations is out of scope, as this is application specific. Systems that employ multiple fan systems, such as those using fan-powered terminal boxes, must be appropriately modeled by the application software to accommodate this scenario.

Example Usage Scenario:

None provided

4.3.1.3.9. Task H - Fan Selection

Task Description:

With the total air flow and pressure requirements (as determined in the preceding steps), in coordination with the engineering design criteria for the fan (i.e., class, type of fan, materials for housing and wheel or blades, etc.), a specific fan may be selected to serve the duct system. Of particular concern regarding fan selection is the acoustical performance of the fan. After all of the fan criteria are established the actual fan selection is typically made using a fan manufacturer’s fan selection program. Performing the actual fan selection is out of scope, as this is application specific.

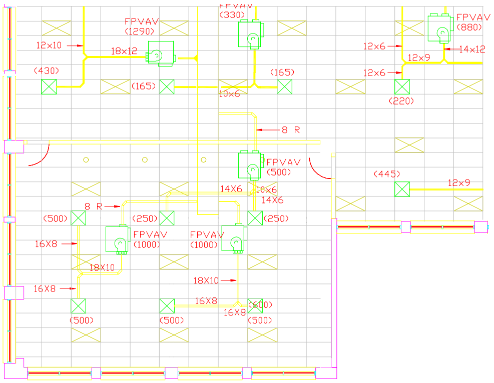
Example Usage Scenario:

None provided

4.3.1.3.10. Task I - Generate Final System Representation

Task Description:

After the components are selected and the duct and fittings are sized the results are used to generate graphical representations showing the actual size and location of the ducts and all of the associated components.



Example Usage Scenario:

None provided

4.3.2. Process: HVAC Piping System Design

4.3.2.1. Introduction

Overview:

See project summary.

Process Scope:

- Select and locate components to be connected in the piping system
- Connect components with pipe and fittings
- Locate other components: strainers, valves, etc.
- Facilitate sizing pipes and fittings
- Facilitate interference checking
- Facilitate pressure drop calculations
- Facilitate pump selection
- Facilitate flow analysis
- Generate final system representation

Out-of-Scope:

- Selection of system type
- Actual sizing the pipe and fittings
- Performing interference checks
- Performing pressure drop calculations
- Performing pump selection
- Performing flow analysis

Definitions:

- ASHRAE - American Society of Heating Refrigeration and Air Conditioning Engineers
- BACnet – Building Automation and Control Network

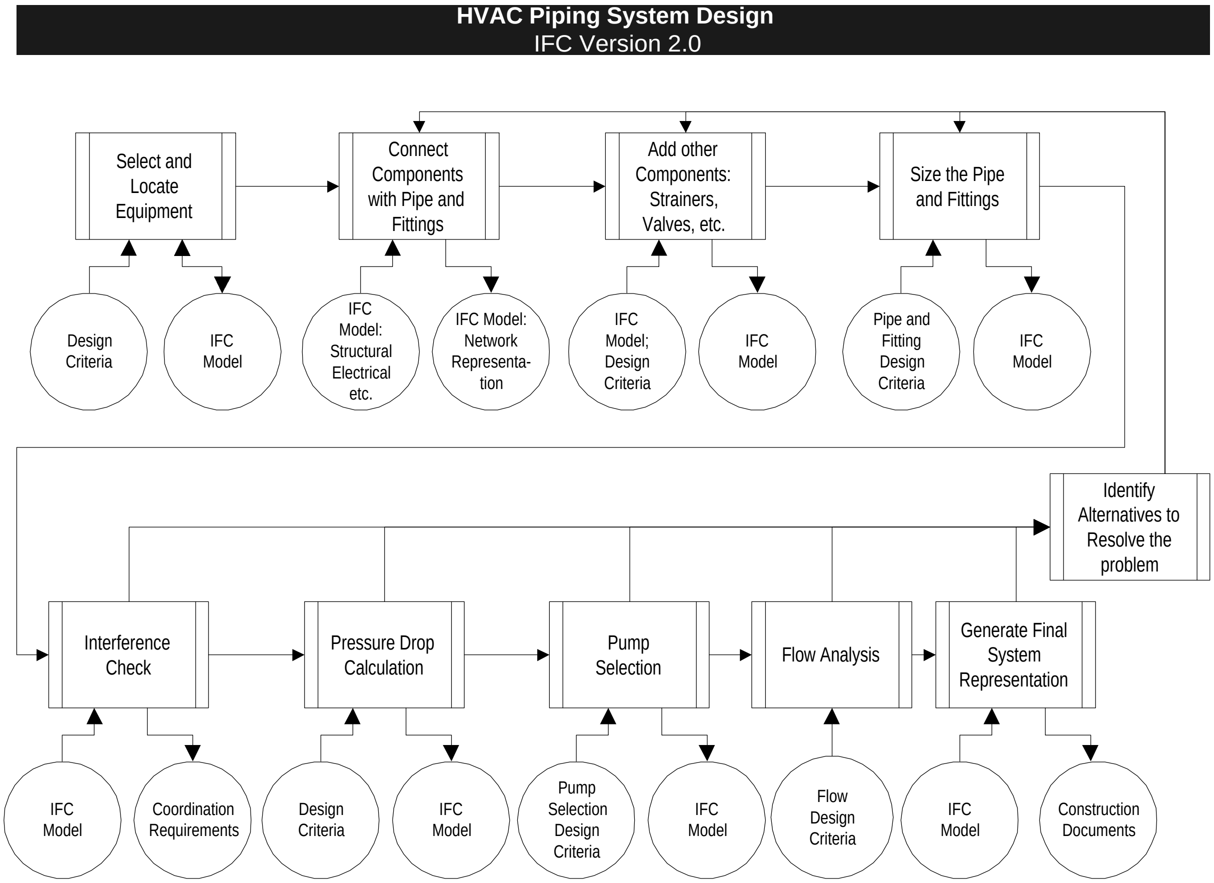
References:

- ASHRAE Handbooks
- BACnet Specifications (ANSI/ASHRAE 135-1995)
- IFC R2.0 BS-3 Project: Pathway Design and Coordination

Contributors:

- James Forester, North American Chapter: Technical Coordinator
- North American Chapter -- Building Services Committee
- United Kingdom Chapter -- Building Services Committee
- German Chapter -- HVAC Committee
- Nordic Chapter -- HVAC Committee
- French Chapter -- HVAC Committee

4.3.2.2. Process Diagram: HVAC Piping System Design



4.3.2.3. Process Definition: HVAC Piping System Design

4.3.2.3.1. Overview

Once an appropriate system type (chilled water, hot water, condenser water, etc.) has been determined (outside of scope), the Piping System Design process begins by selecting and locating pumps, coils, chillers, boilers, heat exchangers, cooling towers, etc., that will be part of the system. To appropriately locate the equipment, architectural and structural information is required so that initial interferences may be avoided.

The next step is to connect the various pieces of equipment together with pipes and fittings, including specifying the types of fittings (i.e., 90 degree elbow, 90 degree long-radius elbow, 45 degree elbow, thru tee, etc.). A graphical representation of this system layout is generated for use in calculating pipe sizes and coordination with other disciplines.

The fluid flow rates, fluid temperature changes, and pressure drops are assigned to the coils, heat exchangers, or other pieces of equipment that remove or add heat or power from the system. The fluid flow rates are determined from the building cooling and heating load calculations (defined in the IFC 1.0 Specifications), the engineer’s design criteria, or from specific equipment requirements.

The pipe and fitting sizes will then be calculated based upon these fluid flow rates and the pipe system design criteria. The pipe and fitting sizes are then reflected in the graphical representation of the system.

Other required system components, such as valves, strainers, etc., are then located on the graphical representation. Any components that require other disciplines to respond are identified, such as electrical or pneumatic power required to operate control valves.

Once these components are located, an interference check is performed. This requires spatial coordination with other building disciplines and may require some pipes to be relocated.

A pressure drop calculation is then performed to determine the system pressure drop. With this information as well as the total fluid flow rate and the engineering design criteria, a pump can be selected.

Primary difficulties in the pipe system design process are coordination with other disciplines to prevent conflicts for space job and to predict sound levels which result from rotating equipment and fluid flow in pipes.

4.3.2.3.2. Task A - Select and Locate System Components

Task Description:

Before the engineer can select and locate system components, the following information is required from other members of the design team:

- Floor plans (architect)
- Reflected ceiling plans (architect)
- Structural plans (structural engineer)
- Duct plans (HVAC engineer)
- Plumbing plans (Plumbing engineer)
- Pipe system design criteria (HVAC engineer)

The selection of equipment (coils, evaporators, condensers, unit heater, radiation, etc.) the piping system will serve is made by the designer, using information from the heating and cooling load calculations in conjunction with manufacturers' equipment information and engineering judgment. The selection of equipment should include the type and size of pipe connections to the equipment. The physical location of the equipment is then determined, giving consideration to space requirements for maintenance and removal of coils and tube bundles.

Example Usage Scenario:

None provided

4.3.2.3.3. Task B - Connect the Components with Pipe and Fittings

Task Description:

Before the engineer can connect equipment with pipes and fittings, the following information is required from other members of the design team:

- Floor plans (architect)
- Structural plans (structural engineer)
- Coordination requirements from any other disciplines

This step involves preparing a schematic representation of the system under design. The various types of elbows, tees and other fittings that are utilized must be designated as to what type is being used and its pressure drop characteristics. These schematic representations are then used to begin coordination with other disciplines that are impacted by the piping system. Interferences must include pipe hangers and supports, and insulation when applicable. Often the piping connections to a given type of coil, unit heater or other piece of equipment are standardized. These assemblies of pipe, valves, fittings, etc., can be treated as standardized piping templates for the given piece of equipment.

Example Usage Scenario:

None provided

4.3.2.3.4. Task C - Locate Other System Components

Task Description:

The locations of other piping system components (i.e., valves, strainers, etc.) are determined from information in the schematic drawings and the design criteria established by the engineer. These components have pressure drops, and connections that may be different from the pipe size. The requirement for some or all of these components may come from equipment selection programs, from standard lists or libraries, or be determined manually by the engineer.

Control elements, such as sensors, actuators and controllers, can also be specified at this point. The design engineer typically defines the general parameters of a control device. However, a control system vendor may utilize many different mechanisms to achieve the desired effect intended by the design engineer. For this reason, only a subset of control element information is necessary.

Example Usage Scenario:

None provided

4.3.2.3.5. Task D - Sizing the Pipe and Fittings

Task Description:

The pipe and fittings are sized using the information derived from the schematic drawing and the design criteria established by the engineer. Design criteria include such things as maximum velocity, pipe material to be used, etc. The actual sizing of the pipe and fittings is out of scope, as this is application specific.

Example Usage Scenario:

None provided

4.3.2.3.6. Task E - Interference Check

Task Description:

Before the engineer can perform an interference check, the following information is required from other members of the design team:

- Floor plans (architect)
- Ceiling grid plans (architect)
- Reflected ceiling and/or lighting plans (architect or lighting designer)
- Power plans (electrical engineer)
- Duct plans (HVAC engineer)
- Plumbing and sprinkler plans (plumbing or fire protection engineer)
- Structural plans (structural engineer)
- Coordination requirements from any other disciplines

Interference checking identifies locations where changes are required in the location of pipes in order to eliminate physical conflicts with other building components or systems. Interference checking must account for insulation, pipe supports and operating and servicing of valves, strainers, etc. For example, placing valves with stems down is not good engineering practice, while a horizontal stem requires more space for the stem, and a vertical stem may require more space at the side for service access. Performing the actual interference check is out of scope, as this is application specific.

Refer also to the IFC R2.0 BS-3 Project: Pathway Design and Coordination.

Example Usage Scenario:

None provided

4.3.2.3.7. Task F - Identify alternatives to design problems

Task Description:

This step requires the designer to go back and redesign certain portions of the system. This may involve regenerating the schematic design documents and recalculating system component sizes. Note that this step may occur at any point in the process.

Example Usage Scenario:

None provided

4.3.2.3.8. Task G - Pressure Drop Calculations

Task Description:

After interference conflicts are corrected the total pressure drop for the system can be calculated. This information is essential to properly select a pump that will serve the piping system. Performing the actual pressure drop calculation is out of scope, as this is application specific.

Example Usage Scenario:

None provided

4.3.2.3.9. Task H - Pump Selection

Task Description:

With the total fluid flow rates and pressure requirements (as determined in the preceding steps) in combination with the engineering design criteria for the pump (i.e., type of pump, pump materials, etc.), the pump selection can be made using a pump manufacturer’s pump selection program. With the selection of the pump, consideration must be given to isolating the pump from the influence of expansion and contraction of the piping system due to temperature changes, and to the transfer of noise and vibration from the pump to the building. Performing the actual pump selection is out of scope, as this is application specific.

Example Usage Scenario:

None provided

4.3.2.3.10. Task I - Flow Analysis

Task Description:

For piping systems with diversified loads (so that all coils do not need maximum flow at the same time, or where there are multiple pumps in the system) the flow rates, pressure drops and temperatures may change randomly. Under these conditions good engineering practice requires further analysis of the flow. The results obtained from the pipe sizing program are necessary to the use of a flow analysis program. Performing the flow analysis is out of scope, as this is application specific.

Example Usage Scenario:

None provided

4.3.2.3.11. Task J - Generate Final System Representation

Task Description:

After the components are selected and the pipe and fittings sized, the results are used to generate graphical representations showing the actual size and location of the pipes, fittings and all of the components.

Example Usage Scenario:

None provided

4.4. [BS-3] Pathway Design and Coordination

Processes Defined in this project:

1. Coordination of mechanical systems
2. Coordination of mechanical systems within the building model

4.4.1. Process: Pathway Design and Coordination

4.4.1.1. Introduction

Overview:

The design of pathways contains the draft layout, the coordination and the representation of mechanical and electrical system-pathways to be installed.

This design process is carried out after the first coordination with the architect and structural engineers, and includes load estimates, energy and systems definitions required for a building.

The process ends with drawings containing the coordinated pathways for the mechanical and electrical installations (i.e. heating, cooling, air-conditioning, plumbing, fire-protection and electrical power) within a building.

The chapter on hand defines the prerequisites for the design of pathway based on generalized design of mechanical facilities.

Process Scope:

- Select and locate plant and other equipment to be connected to the system
- define pathway for different media (duct- and pipe work)
- coordinate pipe- and duct work within the pathway
- coordinate pathways within architectural and structural restraints

Out-of-Scope:

- None provided

Definitions:

- ISO
- DIN
- VDI
- SIA
- ASHRAE
- CIBSE

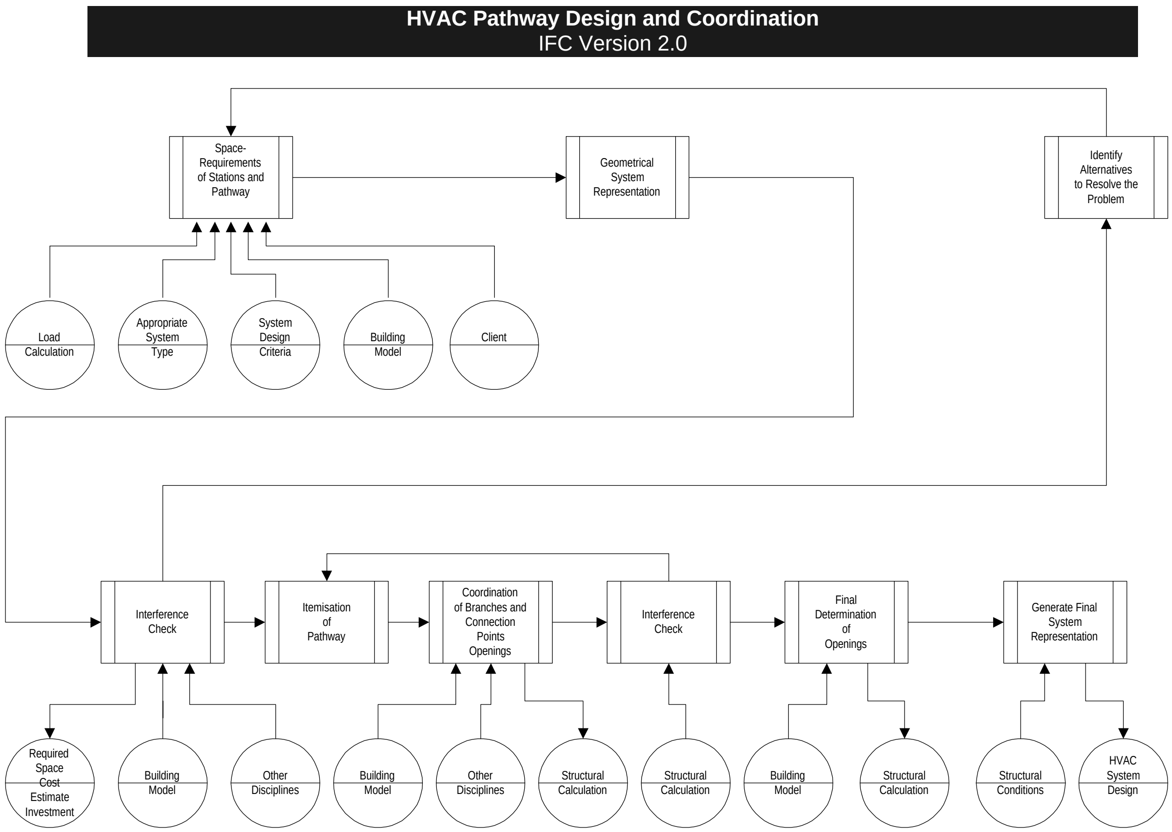
References:

- None provided

Contributors:

- Project Team (see project summary)

4.4.1.2. Process Diagram: Pathway Design and Coordination



4.4.1.3. Process Definition: Pathway Design and Coordination

4.4.1.3.1. Overview

Based on the building model and the conditions (program) defined by the customer, an initial estimate of required energy, technical equipment and systems is defined. The process of designing the pathway starts by defining the required spatial extents for technical equipment, piping, ducting and electrical routes.

A rough building layout by the architect will frequently be available showing the suggested locations for plant rooms and risers.

Considering these parameters, the engineer defines the necessary locations for plant areas and suggests the routing of the main pathways.

The required plant area and main pathways are represented in the M & E drawings.

This draft is presented to the architect/customer with details on space requirements (sections). Thereafter, a review of the suggested design solution will take place, taking into account the structure, the initial and future investment, user requirements, operating expenses and the flexibility achieved.

Parameters from the building model, the definition of systems and the routes of each media type can be combined to define the pathway. Air ducts, including equipment (fire dampers, VAV-boxes, etc.) are combined to form a ventilation pathway. Pipes for heating, cooling or plumbing are combined to form a media pathway. Electrical trays are combined to form an electrical pathway. Each pathway should allow variables for necessary insulation or fire proofing, as well as variables for necessary access for installation and maintenance. The optimization of the pathway itself can be done by varying the distance and position of ducts, pipes or trays. Every pathway must be coordinated within the architectural and structural restraints, as well as with each other.

A final definition of the spatial requirements for technical equipment and media distribution, defines the location of the pathway. The translation of the pathway into geometrical forms is carried out. These drawings serve as a guideline for the ongoing building services design.

The definitions of the structural systems (flat slab, concrete or steel construction, beams, etc.) reflect the location of the plant areas, risers and pathways. Collision detection with walls, slabs, binding beams etc. should be made and openings have to be defined.

4.4.1.3.2. Task A - Defining required space for stations

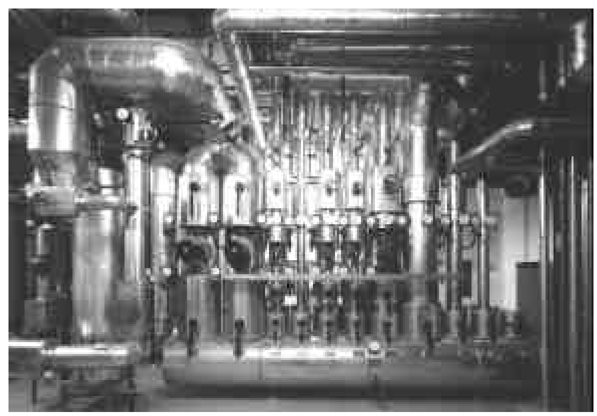
Task Description:

This step contains the dimensioning of main components for different systems, inquiry of the approximate space requirements and corresponding placing of the technical areas in the building model.

Example Usage Scenario:

None provided

4.4.1.3.3. Task B - Defining the required space for pathways



Task Description:

This step contains the dimensioning of the energy and media supply as well as the specification of the pathway.

Example Usage Scenario:

None provided

4.4.1.3.4. Task C - Geometrical representation of stations and pathways

Task Description:

This step contains the geometrical representation of the defined centralized media supply and pathway.

Example Usage Scenario:

None provided

4.4.1.3.5. Task D - Interference check

Task Description:

Collision detection within building services and building model.

Example Usage Scenario:

None provided

4.4.1.3.6. Task E - Identify alternatives to resolve the collisions

Task Description:

This step requires the designer amend and redesign certain portions of the system to resolve possible collisions. This may involve regenerating schematic design documents and recalculating system component and sizes. Note that this step may occur at any point in the process.

Example Usage Scenario:

None provided

4.4.1.3.7. Task F - Itemization of Pathway

Task Description:

This step contains the detailed output of a pathway. Consideration of connection points and branches as well as placement and distance of each pipe or duct, the cross-sectional dimension of the pathway is brought into line with the respective conditions and will be optimized.

Example Usage Scenario:

None provided

4.4.1.3.8. Task G - Coordination of branches

Task Description:

This step contains the coordination of different trades within the design of pathway at branches as well as the coordination with structural conditions like binding beams etc.

Example Usage Scenario:

None provided

4.4.1.3.9. Task H - Interference check

Task Description:

Final collision detection within building services and building model.

Example Usage Scenario:

None provided

4.4.1.3.10. Task I - Determination of openings

Task Description:

This step contains the specification of openings defined by the itemized pathway or single pipes or ducts.

Example Usage Scenario:

None provided

4.4.1.3.11. Task J - Generate final system representation

Task Description:

This step takes us back to HVAC System Design.

Example Usage Scenario:

None provided

4.5. [BS-4] HVAC Loads Calculation

Processes Defined in this project:

1. Building Heating and Cooling Load Calculation

4.5.1. Process: Building Heating and Cooling Load Calculation

4.5.1.1. Introduction

Overview:

Load calculations serve as the basis for all design stages of the building services design. The results of the load calculations enable the designer to dimension the plant equipment and to determine the required space for plant room.

Load calculations are an official proofing method (in Germany for example the proof for heat loss protection must be given in the course of a project), a mode for calculating the heating/cooling load or for the yearly dynamic load simulation:

The process terminates in the complete calculations and the data exchange into the IFC model.

The chapter on hand defines the prerequisites for the computer-aided load calculation .

Process Scope:

- Calculating heating/cooling load

Out-of-Scope:

- bounding conditions like adjacent buildings,

Definitions:

- DIN
- ISO
- VDI
- ASHRAE

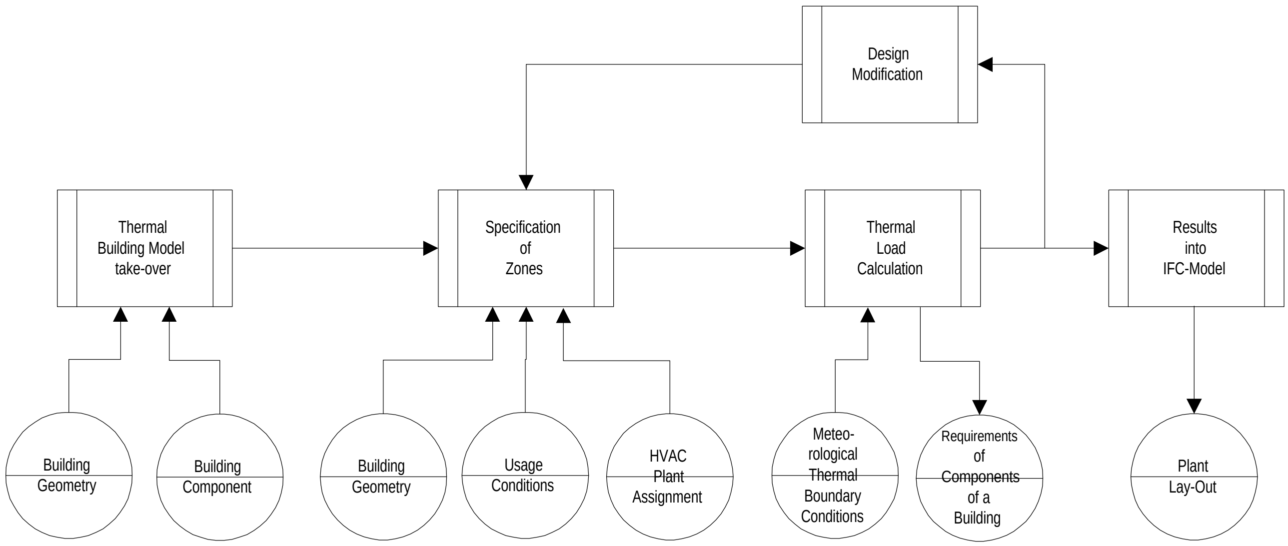
References:

- DIN 4701, 4108
- VDI 6021
- VDI 2078

Contributors:

- Felix Brückner vögtlin engineering
- Rainer HirschbergGTS
- Doris Huber ‘ESS
- Eberhard Michaelis Softtech
- Ulrich Paar Ziegler Informatics
- Robert Rottermann RoCAD Informatik
- Willi Spiegel Triplan GmbH
- Rolf Tonke Planungsgruppe M+M AG
- Kurt Weber Pit-cup GMBH
- Michael Kuehn jr. Kuehn Bauer Partner

4.5.1.2. Process Diagram: Building Heating and Cooling Load Calculation



4.5.1.3. Process Definition: Building Heating and Cooling Load Calculation

4.5.1.3.1. Overview

After the completion of the building model with its geometric and physical building specifications by the architect, the data is to be extracted using the IFC-Model. The IFC-Model includes all architectural building components of a defined room, the attributes and the relationships of the components to each other. The IFC-Model does not include any the description of the adjacent buildings (e.g. input for external shading).

The parameters like the room temperatures, required air changes, people or machine loads or other necessary data is submitted if known to the design team. If certain data is not know to the design team plausible data is assumed to provide preliminary answers.

National boundary conditions has to be transmitted alternatively or in respectively conformist form.

The data exchange to the Calculation-Software does not require any exchange of graphical data. The exchange should be independent from the calculation method applied because it describes only the physical data.

After the exchange of data, the engineer checks data transmitted for completeness and possibly amend the data. The engineer has to input the boundary conditions as well as the meteorological data for the load calculation method.

The definition of zones, as a result of the assigned plant equipment, can be carried out by simply numbering them. All rooms of one level having common boundaries can be defined as one zone. Another form of zoning can be made by direct plant assignment. This method ensures, that considerations of energy as well as the simultaneity of use conditions within plants are considered.

As a results of load calculations, the physical qualities of building components may be changed and submitted to an optimization process. This is requested to the IFC-building model. After changing the corresponding data a further exchange of basic data is carried out and the process starts once more.

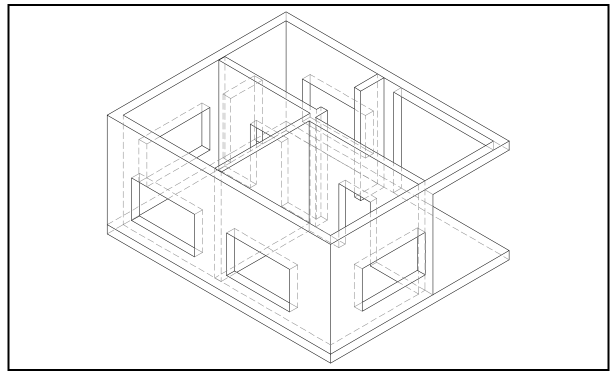
A revision phase is necessary if there is change to the plant assignment or there are variations to the boundary conditions within the process.

At the end of the process the results of the load calculations are provided for the IFC model for further processing. The definition of technical stations, pathway and their space requirements as well as the dimensioning of system components for building services design are based on these results.

4.5.1.3.2. Task A - IFC-Model take-over

Task Description:

This step contains the import of extracted data from the building model like component geometry and component qualities. The construction of this physical data exchange format corresponds in the construction to a Physical-STEP file.



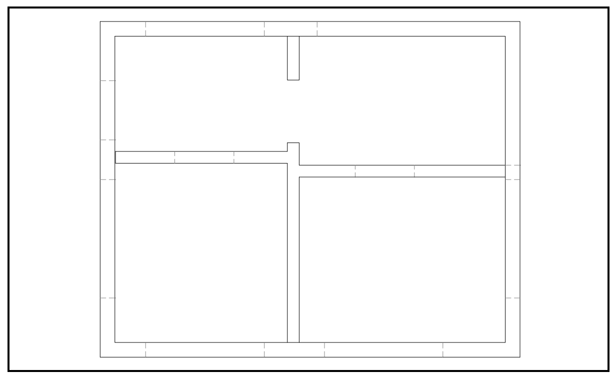
Example Usage Scenario:

None provided

4.5.1.3.3. Task B - Specification of zones

Task Description:

Considering the building geometry, zones are defined for the execution of the load calculations.



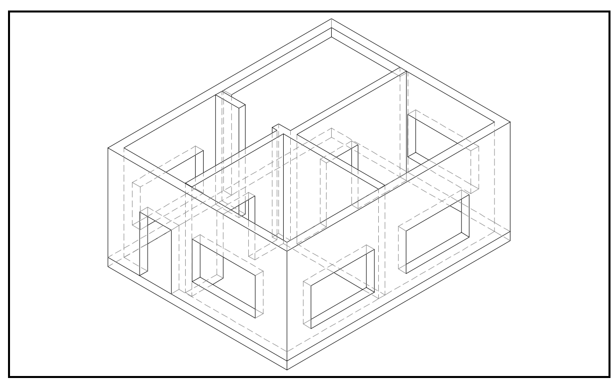
Example Usage Scenario:

None provided

4.5.1.3.4. Task C - Load calculations

Task Description:

This step contains the execution of the load calculations.



Example Usage Scenario:

None provided

4.5.1.3.5. Task D - Results into IFC-Model

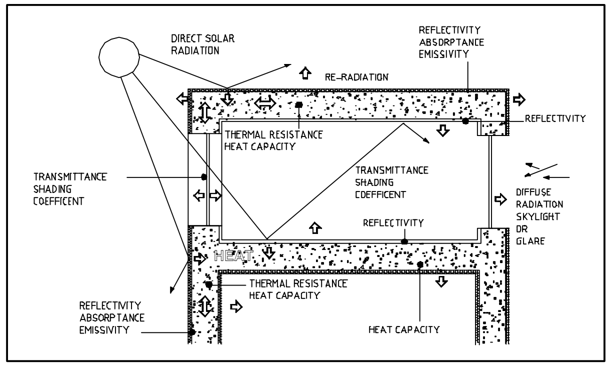
Task Description:

Exchanging the results of the load calculations to the IFC model.

Example Usage Scenario:

None provided

4.5.1.3.6. Task E - Design modifications



Task Description:

This step contains the iterative event for the execution of all calculations by variation or change of the zones, usage requirements etc., according to optimization by changing parameters .

Example Usage Scenario:

None provided

4.6. [CS-1] Code Checking - Energy Codes

Processes Defined in this project:

1. Commercial and Residential Energy Code Compliance Checking

Code compliance is performed by building designers, systems designers, and code enforcement officials. Compliance with codes begins during programming when designers determine which codes apply to the building project. Preliminary code reviews are frequently performed during schematic design and more thorough reviews are performed by members of the design team late in the design process before construction documents are complete. Building code officials perform plan reviews as part of the building permitting process. Designers and code officials perform drawing takeoffs as necessary to ensure compliance. Information about building systems, assemblies, layout, etc. is gathered during this process and compared to the requirements for each applicable code. Virtually all systems within a building are constrained in some way by codes (or voluntary design standards), hence codes are relevant to most other design processes. Energy codes, the subject of this Release 2.0 proposal, are strongly related to architectural, HVAC, and electrical design processes.

Code compliance checking is the process of assessing whether a building complies with codes enforced by local jurisdictions or with voluntary design standards promulgating by various standard-writing entities.

4.6.1. Process: Commercial and Residential Energy Code Compliance Checking

4.6.1.1. Introduction

Overview:

This process will support applications that determine whether buildings conform with energy-efficiency codes for new construction. The CS-1 project will focus on two model codes that are widely used in the United States. The project will primarily address requirements pertinent to building envelope and lighting.

Process Scope:

- Commercial energy code compliance (e.g., ASHRAE/IES 90.1-1989 [Code])
- Residential energy code compliance (e.g., MEC)
- Prescriptive code requirements
- Performance code requirements

Out-of-Scope:

- Determination of which codes apply
- Modeling of code requirements (i.e., the object model will not include the code requirements)
- Modeling of energy code provisions not normally addressed on the building plans; e.g. compliance procedures, detailed product and construction specifications, and other information normally relegated to project specifications.

Definitions:

- MEC: Model Energy Code
- HVAC: heating, ventilating, and air-conditioning

References:

- Model Energy Code, The Council of American Building Officials; Falls Church, VA; 1993.
- ASHRAE/IES Standard 90.1-1989, Energy Efficient Design of New Buildings Except Low-Rise Residential Buildings; ASHRAE, Atlanta, GA; 1989.
- Energy Code for Commercial and High-Rise Residential Buildings, Codification of ASHRAE/IES 90.1-1989 Energy Efficient Design of New Buildings Except Low-Rise Residential Buildings; ASHRAE, Atlanta, GA; 1993.

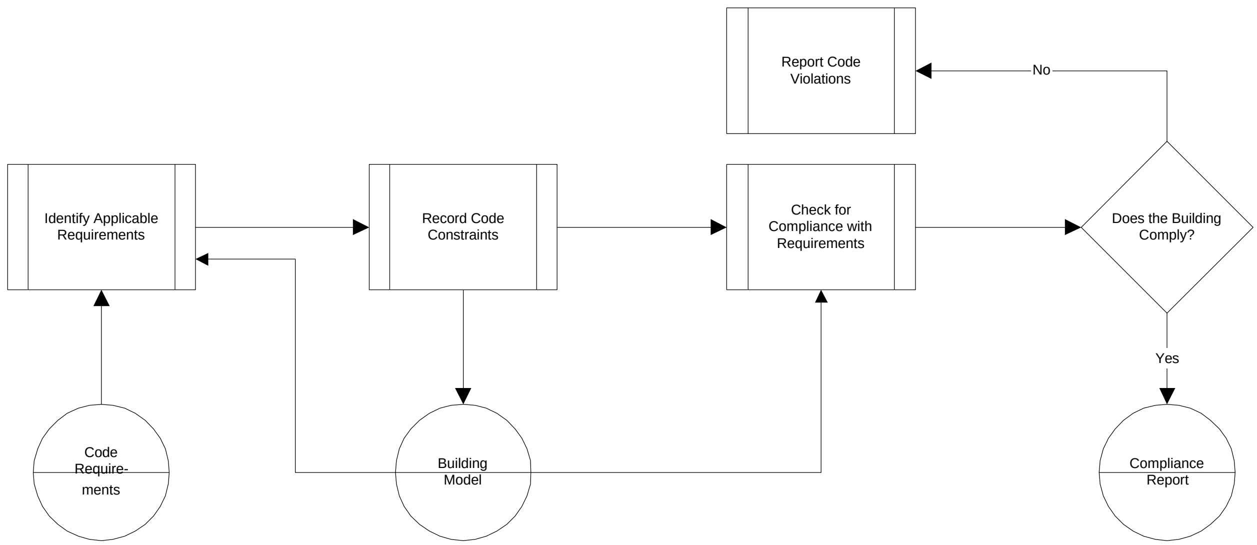
Contributors:

- Rob Briggs, PNNL (NA)
- Tan You Tong, ITI (S)
- Tan Kee Wee, NCB (S)
- Philippe Debras, CSTB (F)
- Robert Amor, BRE (UK)
- Dave Chassin, PNNL (NA)
- Filiz Ozel, ASU (NA)
- Han Kiliccote, CMU (NA)

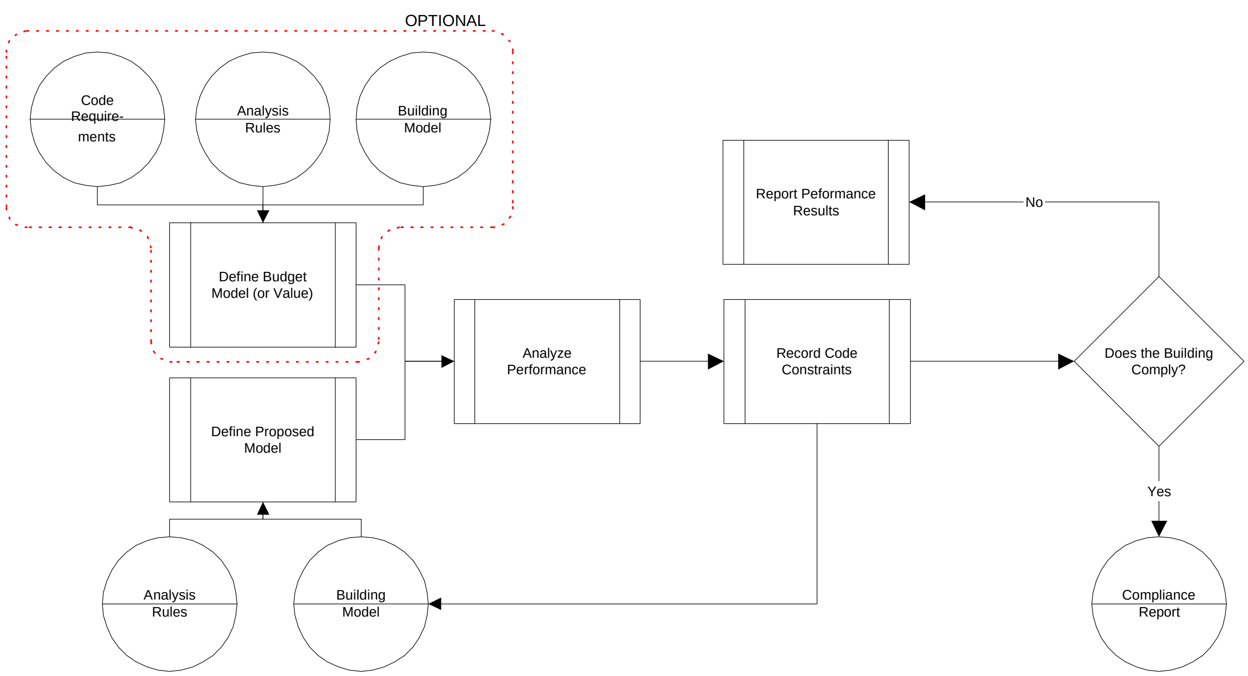
4.6.1.2. Process Diagram: Commercial and Residential Energy Code Compliance Checking

Note: Building codes typically employ two different regulatory approaches: prescriptive requirements and performance requirements. The two-part diagram below illustrates the two different processes corresponding with these two different approaches. Most codes are neither purely prescriptive nor purely performance-based, but rather, contain elements of both types of requirements either in combination or as alternative paths for demonstrating compliance. In its simplest form, a prescriptive requirement says that object or attribute A must have value B. In contrast a performance requirement says object A must perform as well as C, where C can range from a model whose performance must be analyzed to a static value for a performance metric.

Part A - Process for Prescriptive Code Requirements



Part B - Process for Performance Code Requirements



4.6.1.3. Process Definition: Commercial and Residential Energy Code Compliance Checking

4.6.1.3.1. Overview

Applicable energy codes are normally identified at the programming stage of the project. At the beginning of schematic design, the architect, HVAC engineer, energy consultant, or other designated design team member with responsibility for energy code compliance identifies those code requirements likely to constrain the building design. Depending on the severity of the code constraints, compliance with these requirements may be spot checked as the design process progresses, or the energy requirements may be largely ignored until a final compliance check is done, usually at the end of the design development phase of the project.

Most energy code requirements are not strictly prescriptive, but rather constrain the performance of an assembly, subsystem, or major building system. Determining compliance with these requirements frequently requires multiple inputs and some computation. Enabling the necessary data to be managed and manipulated using IFC's will eliminate manual tasks and enable energy code compliance to be checked more easily and frequently during the design process, resulting in compliance at lower cost and with less disruption to the design process. The capability to associate code constraints with objects in the building model will enable design applications to monitor conformance with codes without concurrent operation of code-checking applications. Designers can then focus on the design with confidence that they will be notified if proposed design changes violate code requirements.

4.6.1.3.2. Task A - Identify Applicable Code Requirements

Task Description:

This process begins with the intent to demonstrate that a given proposed building design complies with a given energy code.

Some requirements in the code (or even major sections of the code) may not apply due to particular characteristics of the project, such as its location, intended use, or number of stories. Some specific energy code-related examples of requirements that are conditionally applicable based on project characteristics are listed below.

- Certain buildings may be exempted from all envelope insulation requirements based on very low connected loads or the absence of space-conditioning equipment.
- Insulation of the exposed perimeter edges of slab-on-grade construction is required in climates with greater than 3,000 heating degree days base 65ºF but is not required in climates with 3,000 or fewer heating degree days.

The applicability of other code requirements may depend on specific conditions or exceptions in the code or on definitions of the objects addressed in the requirements. These conditions must be evaluated before the relationship between a building object and an applicable code constraint can be established. Some examples of these conditions are listed below.

- Exterior above-grade walls are subject to insulation requirements, but parapet walls and wing walls are exempt from these requirements.
- Insulation requirements apply to interior walls separating conditioned from unconditioned spaces but otherwise do not apply to interior walls.
- Basement wall insulation is required in many locations, but it is not required when walls are more than one story below grade.
- Either wall or roof insulation requirements may apply to steeply sloping roofs depending on the slope of the assembly.
- Lighting efficiency requirements apply for most building use types, but they do not apply to hotel guest rooms.

Example Usage Scenario:

Figure 1 shows an insulated slab edge. The applicability of code requirements governing the R-value and depth of this insulation is dependent on the climate in which the building is built and whether or not the slab edge occurs at the boundary between conditioned and unconditioned space. If the location has 3,000 or fewer heating degree days base 65°F or if the space circumscribed by the slab perimeter is unconditioned, no slab edge insulation is required.

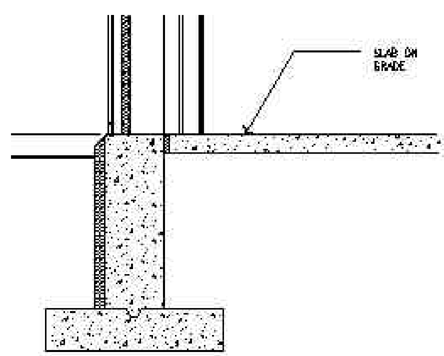


Figure 1 - Insulated Slab-on-Grade Perimeter

Figure 2 shows a parapet wall. Although on the building elevation the parapet wall may be indistinguishable from the exterior wall, the insulation requirement that applies to the exterior wall does not apply to the parapet wall.

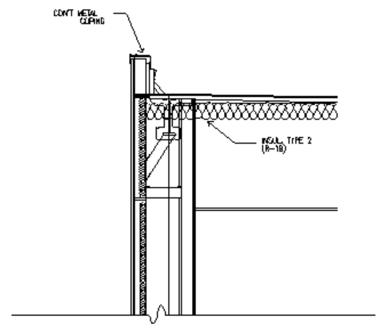


Figure 2 - Parapet Wall

4.6.1.3.3. Task B - Record Code Constraints

Task Description:

Where code requirements constrain the building design, it may be useful to record the code constraint for future use by other applications. The value of storing code constraint information in the building model (as opposed to simply reporting a compliance result) is that it can provide persistent guidance to the user and enable user notification when design modifications are made that will affect compliance. Prescriptive code constraints can be represented as discrete limiting values, which can be associated with a building object and stored in the building model for other applications to utilize and to document the basis for design decisions.

In order for this constraint object to be fully useful, it needs to carry the following information:

- The object to which the constraint is connected
- The numeric and logical content of the constraint
- Identification of the code to which the constraint belongs
- Identification of the application that established the constraint
- A description of the constraint
- Text to be used in notifying the user about the constraint
- Other objects and attributes on which the value or application of the constraint depends.

Unlike with prescriptive requirements, a performance-based requirement cannot be expressed as a discrete constraint on an individual object. Rather, the constraint is typically imposed on a system consisting of multiple objects that interact within the code-constrained system. To accommodate performance-based code constraints, it is necessary to attach the constraints to aggregate objects. In addition, many, though not all, performance codes employ requirements that are not fixed values but rather are themselves the results of calculations. These requirements tend to have a larger number of dependencies on other objects in the model, and hence are more likely to be affected by other design changes.

Example Usage Scenario:

Figure 3 shows a window assembly consisting of a window and a window frame. The U-factor of the window assembly--not the glass or the frame but the combined assembly--is constrained by a code requirement. For purposes of this example, the code constraint (from ASHRAE/IES Standard 90.1-1989) is that the U-factor of the window assembly may not exceed 0.72 Btu/($h \cdot ft^2 \cdot °F$). Listed below is the information that would be recorded with the code constraint and that would be available to other applications.

Code constraint attached to aggregate object *Window Assembly*.

- Numeric and Logical Content: Window assembly U-factor must be less than or equal to 0.72 Btu/(h·ft^2·°F).
- Identification of Code and Section Number: ASHRAE/IES Standard 90.1-1989, Section 8.6.10.2(b).
- Constraint Established By: COM*check-EZ*, Version 2.0.
- Description of Constraint: The code requires that locations with greater than 3,000 heating degree days base 65°F have an overall U-factor (i.e., including both glass and frame) that does not exceed 0.72 Btu/(h·ft^2·°F).
- Text for User Notification: "The overall U-factor (i.e., including both glass and frame) exceeds 0.72 Btu/(h·ft^2·°F) and therefore violates ASHRAE/IES Standard 90.1-1989, Section 8.6.10.2(b)."

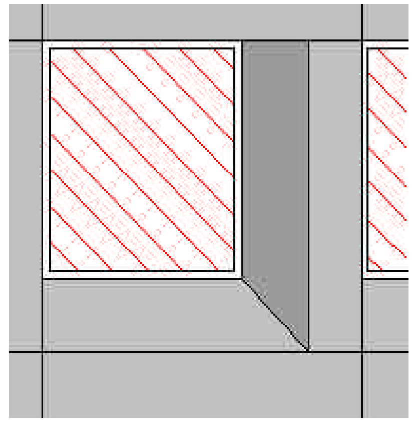


Figure 3 - Window and Frame

This Constraint Depends On: 1) site heating degree days base 65°F, 2) space conditioning of parent space, 3) glass U-factor, 4) glass area, 5) frame U-factor, and 6) frame

4.6.1.3.4. Task C - Check for Compliance with Prescriptive Requirements

Task Description:

This second step for compliance checking with prescriptive code requirements involves a logical comparison of the applicable prescriptive requirements in the code with the corresponding objects and attributes in the building model. This checking process yields both a status result and a code constraint on each of the corresponding building attributes. Commercial energy codes usually contain requirements that pertain to the architectural envelope, lighting systems, and HVAC and service water heating systems. Residential energy codes usually address only building envelope, HVAC, and water heating.

Example Usage Scenario:

Figure 1 shows the perimeter of a concrete slab on grade that has been insulated using vertically placed insulation that extends downward 24" from the top of the slab. In Minneapolis, this insulation must have an R-value of 8 or greater. The compliance checking process for prescriptive requirements simply involves executing logical comparisons between the applicable code requirements and the corresponding attribute(s) of code-constrained objects.

4.6.1.3.5. Task D - Define Budget Model

Task Description:

Compliance checking with performance-based requirements frequently requires that three steps be taken that are not required with prescriptive requirements: defining a budget building model, defining a proposed building model, and analyzing the performance of each. Defining the budget model (i.e., the model or building configuration that defines code-minimum performance) is typically performed by implementing prescriptive code requirements into a copy of the description of the proposed design. For example, the code checking procedure may substitute the prescriptive wall and roof insulation requirements for those used in the proposed design. Other assumptions may be imposed to ensure a fair basis for comparison with the performance results from the proposed model; for example, by specifying consistent operating assumptions and energy prices. In addition to implementing these modifications, this step involves translating the representation in the building model to the appropriate representation and format required by the simulation model used to analyze performance.

However, as indicated in the process diagram, a common variation for performance code requirements is to have the budget value be a static metric of performance. In this case, the process simplifies to a logical comparison of performance values similar to the prescriptive compliance check.

Example Usage Scenario:

Figure 4 shows a building floor plan with three different space usage (or task area) designations. The lighting sections of commercial building energy codes set lighting power budgets for various types of spaces based on their usage. The lighting power budget is generated by multiplying the area of each space type by its permitted lighting power density. This approach is performance-based because the resulting budget is applied at the whole-building level, and users are free to use any combination of lighting fixture types and quantities provided the aggregate input wattage does not exceed the budget.

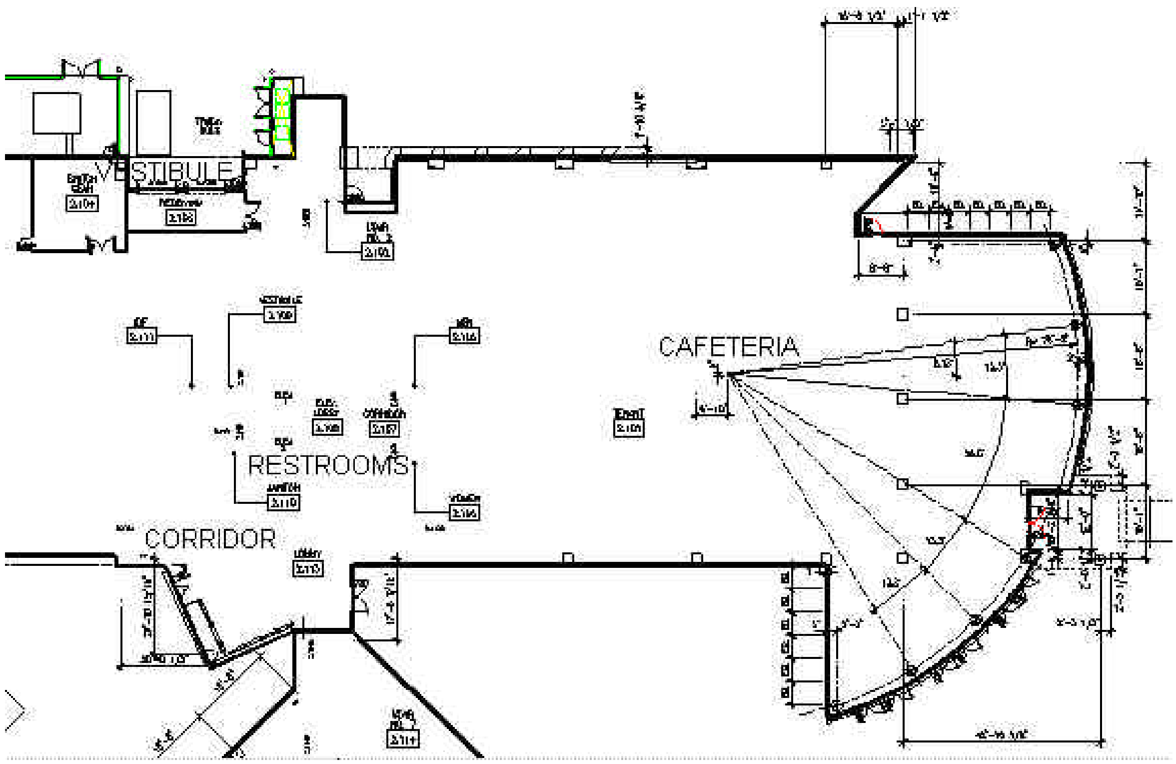


Figure 4 - Floor Plan

The actual values that make up the lighting budget model are shown in Table 1.

Table 1 - Example Budget Model for Performance-Based Lighting Compliance

Space Type Designation	**Floor Area (ft^2)**	**Lighting Power Density (W/ft^2)**	**Lighting Power Budget (W)**
Cafeteria	4,400	2.5	11,000
Restrooms	300	0.8	240
Vestibule	800	1.0	800
Corridor/Stairs	1,000	0.8	800
Total	6,500	---	12,840

4.6.1.3.6. Task E - Define Proposed Model

Task Description:

A similar process is used to define the proposed model as was used to define the budget model. Most objects in this model are defined directly from the building model entered by the user, however some assumptions may be imposed to ensure a fair comparisons between budget and proposed models. A similar translation is made to the required format for the simulation model.

Example Usage Scenario:

Figure 5 shows the reflected ceiling plan for the same areas shown in Figure 4. For lighting compliance, the fixture descriptions, quantities, and input wattages that are specified for the building are used to define the proposed model. Table 2 lists the parameter values for the system shown on the reflected ceiling plan (Figure 5).

[Add Reflected Ceiling Plan Here]

Figure 5 - Reflected Ceiling Plan

Table 2 - Proposed Design Model for Performance-Based Lighting Compliance

Fixture Description	Quantity	Input Wattage (W)	Design Lighting Power
Total			

4.6.1.3.7. Task F - Analyze Performance

Task Description:

The third step with performance code requirements is to calculate the performance of the two models, one representing the design proposed by the user and the other establishing code-minimum performance. While this process description implies that a the analysis involves a complex process, the basic sequence is applicable to a wide range of analysis procedures. The required calculation may be as simple as a parallel path calculation that combines the rated thermal conductances of two components to determine an overall assembly conductance (as in Figure 3 for the window assembly), or it may be as complex as an annual computer simulation of building energy use for a multi-zone building.

Example Usage Scenario:

Table 1 and Table 2 show the calculation of lighting power budget and the connected load for the proposed design. In this lighting compliance example, the performance analysis involves multiplying values in each of the rows and summing the products in the right-most columns.

4.6.1.3.8. Task G - Compliance Determination and Reporting

Task Description:

Depending on the compliance outcome from the code checking sequence, the user is either notified of the code violation or notified that the design complies and given the opportunity to generate a compliance report for submission as part of the building permit application. When a prescriptive code requirement is violated, the specific features that violate the requirement are listed for the user along with the corresponding code constraints. Such notification normally prompts the user to modify the design and rerun the compliance check to document compliance. The user may also leave the design unchanged and demonstrate compliance using an alternative, performance-based compliance method.

Unlike with prescriptive requirements, a failure to comply with a performance-based requirement cannot be attributed to the violation of a specific requirement but may depend on a large number of building objects. Results from a performance code evaluation are reported in the form of performance relative to a performance budget. This fact often leads the user to an iterative process to resolve the code violation. The user evaluates various ways of achieving compliance, and often a variety of design and cost issues are considered before a design change is accepted.

Example Usage Scenario:

Figure 6 shows a portion of a compliance report from an energy code application that has both prescriptive and performance-based requirements. Note that at the bottom of the page, the report indicates that the building complies by a certain percentage with the performance requirements of the code but that it also notes the violation of a specific prescriptive requirement. In this case, the example building will not comply until the violation of the prescriptive requirement is corrected.

ENVELOPE COMPLIANCE CERTIFICATE
COMcheck-EZ Software Version 2.0

Permit #

Section 1: PROJECT INFORMATION

Chkd by/Date

Project Information:
ABC Stores Inc.
1234 5th Ave.
George, Washington 98765

Designer/Contractor Information:
ENR Design and Construction Inc.
567 George Washington Way
Martha, Washington 98766

CLIMATE-SPECIFIC REQUIREMENTS

Component Name/Description		**Area**	**Cavity R-Value**	**Cont. R-Value**	**Assembly U-Factor**	**Budget U-Factor**
ROOF: Nonwood Joist/Truss		4500	0.0	16.0	0.060	0.057
WALL: Metal Frame, 16" o.c.		600	13.0	0.0	0.132	0.086
WALL: CMU >8"/Int. Ins/Mtl	1200	13.0	0.0	0.112	0.086	
WALL: CMU >8"/Int. Ins	680	--	0.0	0.273	0.086	
WIN: Low-E/Clr/Tbrk/SHGC=0.71/PF		500	--	--	0.570	--
WARNING: The following window violates a mandatory U-factor requirement						
WIN: Single/Clr/Mtl/SHGC=0.87/PF		100	--	--	1.170	--
DOOR: Opaque	35	--	--	0.700	--	
DOOR: Glass	40	--	--	0.920	--	
SLAB: Unheated w/36" Vertical		280	--	6.0	--	--

Envelope PASSES Performance Requirements: Design 2% better than code.

Figure 6 **- Example Energy Code Compliance Report**

4.7. [CS-2] Code Checking Extensions

Processes Defined in this project:

1. Code Compliance Checking for Disable Access and Escape Route of Commercial, Residential and Institutional Buildings

4.7.1. Commercial, Residential and Institutional Code Compliance Checking for Disable Access and Escape Route

4.7.1.1. Introduction

Overview:

This process will support applications that determine whether building design compile with building codes related to disable access and escape route. CS-2 project will based on codes used in Singapore. The project will address requirements related to usage, dimension, material and relationship of building spaces.

Disable access code compliance process check whether the access provisions and facilities of a building complies with one or more codes or standards that serve the needs of the wheelchair user enforced by various codes and standards promulgation entities. The provisions also apply to ambulant disabled.

Escape route code compliance process check whether the exit provisions and facilities of a building complies with one or more codes or standards that provide safe means of escape for occupants enforced by various codes and standards promulgation entities.

Process Scope:

- Check code compliance for commercial buildings of public resort and concourse (e.g. banks, concert halls, cinemas, hotels, religious buildings, theatres and stadiums).
- Check only the interior of building.
- Check prescriptive based code requirements.
- Check disabled access
  - *Check clearance (width, area for wheelchair movement) of space and access*
  - *Check surface requirement (materials) of space and access*
  - *Check floor level changes (gradient) of space and access*
  - *Check obstacles on space and access*
  - *Check access aids provision (symbol, handrails)*
- Check escape route
  - *Check occupancy capacity requirements*
  - *Check exit requirements (number, location, capacity and minimum width of exits)*
  - *Check approach to exit (smoke free, fire resistance, sprinkler protected)*
  - *Check area of refuge (smoke free, fire resistance, sprinkler protected)*
  - *Check means of escape (staircase and passageway)*
  - *Check escape aids provision (symbol)*

Note: The above checks are based on the usage and floor area of a space, story, zone or building

Out-of-Scope:

- Check performance based code requirements
- Exterior space
- Check disabled access
  - *Check sanitary provisions (washroom, water closet, basin, urinals, bath)*
  - *Check circulation provisions (lift, conveyance)*
  - *Check transport provisions (car park)*
- Check escape route
  - *Check travel distance*
  - *Check escape requirements for specific type of building (e.g. hotels, hospitals)*

Definitions:

For disabled access:

- Accessible Route: a continuous unobstructed path connecting all accessible elements and spaces in a building that can be maneuvered by people with physical disabilities.
- Disable Person: A person who is wheelchair bound which affects his mobility (only in CS-2 context).

References:

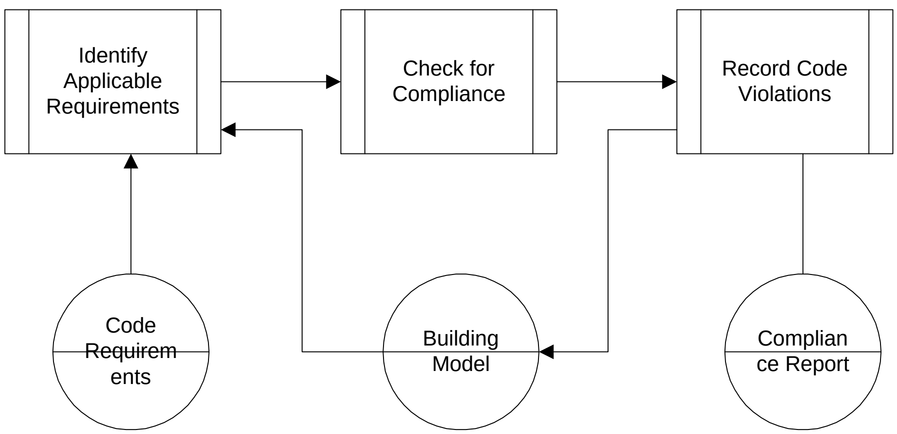
- Code of Practice on Barrier-free Accessibility in Building 1995, Building Control Division, Public Work Department, Ministry of national Development, Singapore.
- Code of Practice for Fire Precaution in Building 1991, Fire Safety Bureau, Singapore Civil Defence Force.

Contributors:

- Wong Wai Ching (S)
- Zhong Qi (S)
- Tan Kee Wee (S)

4.7.1.2. Process Diagram

Note: The process diagram found below is adopted from that of CS-1. This is to ensure consistency between CS-1 and CS-2. As the scope of this project only cover the prescriptive approach to building codes compliance hence only the process for prescriptive code requirements is given below.



4.7.1.3. Process Definition

4.7.1.3.1. Overview

The process is used by building designers and code enforcement officers to ensure code compliance during design and submission stage, respectively. With the process, designers can detect code violations in their design as early as possible while design changes are still relatively cheap to make. Similarly, code enforcement officers can verify the plans submitted by the designers for building approvals faster and more objectively. As both the professionals are using the same checking system, consistency in code interpretation is automatically ensured.

A plan checking system must be able to retrieve information from the design and to be checked against the building code. It may perform more computations to derive additional information from the retrieved before checking is possible. This is because the derivable information is not usually found in the building model in order to keep the size of the exchange model small.

4.7.1.3.2. Task A - Identify Applicable Code Requirements

Task Description:

This process involves identifying code requirements to be checked from a code requirement knowledge base. It should allow user to invoke either all or certain part of the requirements captured in the knowledge base.

Specific code requirements to be complied with will be generated from this task which in turn loads the import routines to retrieve specific building information from the building model to be checked.

Example Usage Scenario:

From an end-user perspective, the task involves mainly deciding which code requirements to be complied with based on the type, usage and occupancy of the building or specific part of the building (either in terms of zone, storey or space). The code requirements can be grouped into the following topics:

- Space clearance and allowance (width, area for wheelchair movement) of space and access
- Floor surface requirement (materials) of space and access
- Floor level changes (gradient) of space and access
- Obstacles on space and access
- Access aids provision (symbol, handrails)
- Occupancy capacity requirements
- Exit requirements (number, location, capacity and minimum width of exits)
- Approach to exit (smoke free, fire resistance, sprinkler protected)
- Area of refuge (smoke free, fire resistance, sprinkler protected)
- Means of escape (staircase and passageway)
- Escape aids provision (symbol)

From a system point of view, a user interface (e.g. in a form of check boxes) should be provided to facilitate the selection of the above requirements. Once the selection is completed, the specific modules of checking routines are then loaded. Next, the user will then be prompted to provide the source of building model. The system can provide an interactively means, e.g. a “rubble bend” polygon, for user to define the extend of the floor plan to be checked or an import facility to read the predefined building model from a physical file or a server. Once the building model is read into the system, instances of building object will be instantiated. From the building objects’ interface, specific attributes of the object can then be retrieved by the system to checking purpose.

4.7.1.3.3. Task B - Check for Compliance

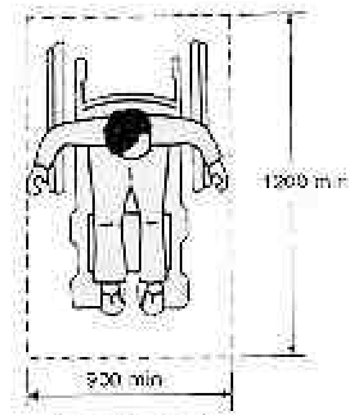
Task Description:

In CS-2, only prescriptive checking is covered. It involves a logical comparison of the applicable prescriptive requirements in the code with the corresponding objects and attributes in the building model. The result of this step is a set of design violations. A series of computation, derivation and code cross-referencing are expected before the comparison can be carried out.

Example Usage Scenario:

From end user perspective, the building will be processed portion by portion until all the requirements are met. A few examples are given below:

- Space clearance and allowance requirements
    - *The minimum clear floor space required to accommodate a single, stationary wheelchair shell be 900 by 1200mm as illustrated in figure 1.*

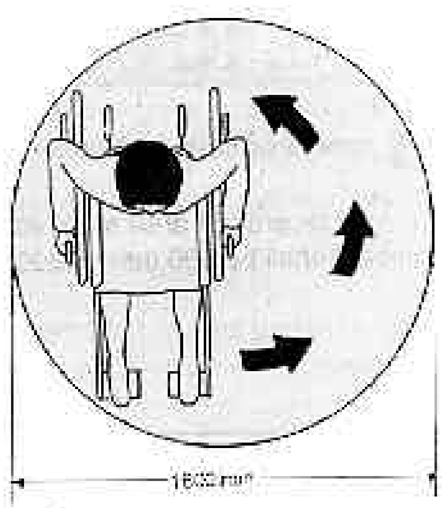


Clear floor space

Figure 1 Minimum Clear Floor Space

- The minimum clear floor space for wheelchair to turn shell be 1800 by 1800mm as illustrated in figure 2.

Figure 2 Minimum Clear Turning Space



Minimum Clear Turning Space

- The minimum clear width of an accessible route shell be 1200mm to allow both wheelchair and a walking person to pass as illustrated in figure 3.

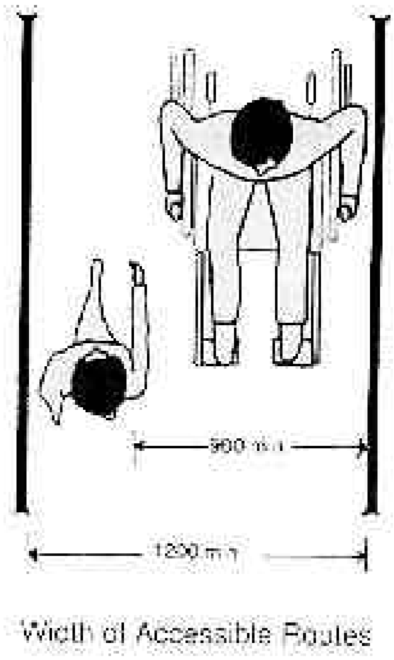


Figure 3 Width of Accessible Routes

- Floor surface (slip resistance) requirements
  - *The slip resistance of a few typical flooring surface*

Material	Slip resistance		Remarks
	Dry & Unpolished	Wet	
Clay Tiles	Very good	Very good	Carborumdum finish
Carpet	Very good	good	Must be securely fixed
Rubber (sheet/tiles)	Very good	Very poor	Not suitable for toilet
…	…	…	…

- Floor level changes requirements
  - *Any changes in level, except for lifts, shell conform to the following table*

Changes in vertical rise (mm)	Gradient not steeper than
0 to 15	1:2
15.1 to 50	1:5
50.1 to 200	1:10
Exceeding 200	1:12

- Obstacles on space and access
  - *Any building element (e.g. column or cabinet) that post as obstacle (leaving insufficient clear width or turning space for the wheelchair) on a disabled accessible route or space shell be avoided.*
  - *Wheelchair manoeuvring at the doorway shell be free of obstruction and be provided with clear space (as illustrated in figure 4) on either side of the door in the following manner:*
    - *For one-way swing door, minimum space of 300mm and 600mm, on the push and pull sides of the door, respectively.*
    - *For two-way swing door, a minimum space of 300 mm.*

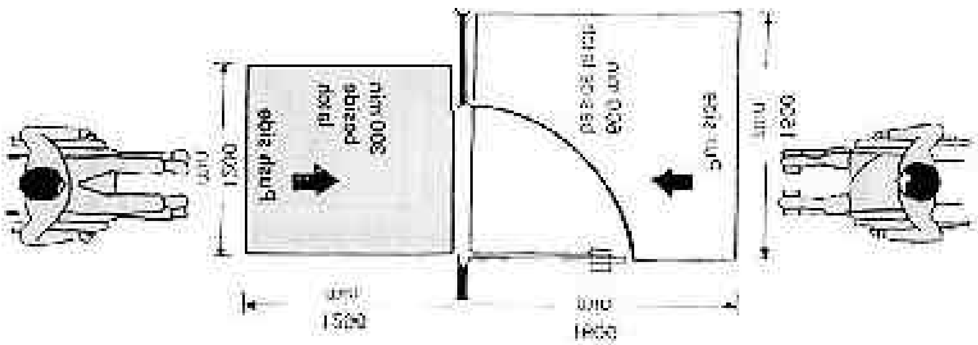


Figure 4 Minimum Doorway manoeuvring Space

- Access aids provision (handrail, ramp, symbol)
  - *Handrail shell he slip resistant, have circular section of 30-45 mm in diameter, continuous gripping surface, without interruptions that can break hand hold and have a clear space of not less than 40 between the handrail and the wall as illustrated in figure 5.*

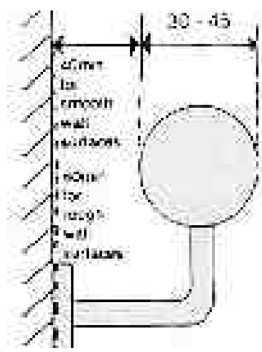


Figure 5 Handrail

  - *Ramp (an inclined way connecting one level to another) shell have a level landing at the top and bottom of each run and also where the run changes in direction as illustrated in figure 6. (The minimum space for landing found in between runs shell be at least 1500x1500mm and that found at the top and bottom run shell be at least 1200x1500mm)*

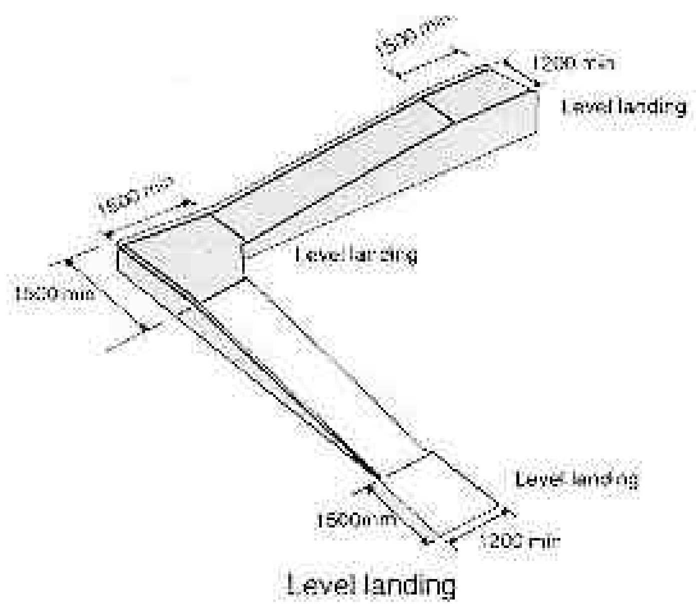


Figure 6 Level Landing

- *Ramp with horizontal run exceeds 9000m shell have a level landing as illustrated in figure 7.*

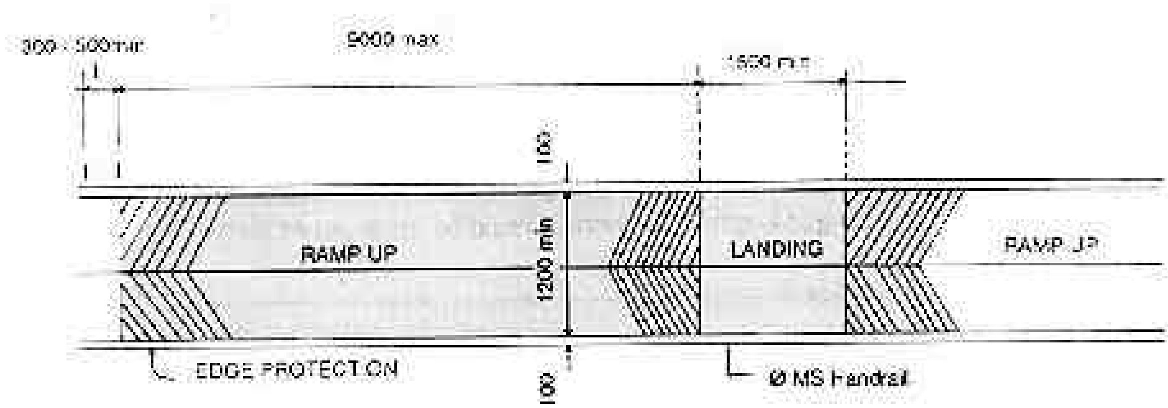


Figure 7 Ramps and Landing

- Occupancy capacity requirements

Type of Occupancy	Capacity (number of persons pre unit width of exit)			
	Exit to external space	Exit to Internal space	Staircase	Corridors, Exit passageway
Commercial (shop, office)	100	80	60	100
Hotel	50	40	30	50
…	…	…	…	…

- Exit requirements (number, location, capacity and minimum width of exits)
  - *Number of exit*

Type of Occupancy	Maximum Occupant with one Exit
Commercial (shop, office)	50
Hospital Ward	15
…	…

  - *Number of exit staircase: at least two independent exit staircases per storey.*
  - *Minimum width*

Type of Occupancy	Minimum Width (m)	
	Door	Stairs
Commercial (shop, office)	1	1
Hotel	1	1
…	…	…

  - *When more than one exit is required from any space, the exits shell be place as romote as possible from the others.*
- Approach to exit (smoke free, fire resistance, sprinkler protected, ventilated and pressurized)
  - *All internal exit passageway shell be naturally ventilated by fixed ventilation openings in external wall, such ventilation openings shell not be less than 15 percent of the floor area of the passageway.*
  - *In any building of which the habitable height exceeds 24 m, any internal staircases without provision of natural ventilation shell be pressurized to comply with the pressurization regulations.*
  - *The floor area of a smoke-stop lobby shell be not less than 3m2 …*
  - *Walls shell have a fire resistance of at least 1 hr.*
- Area of refuge (smoke free, fire resistance, sprinkler protected, ventilated and pressurized)
  - *Exit doors between Area of refuge and external space shall have fire resistance of at least half an hour and fitted with automatic self-closing device to comply with requirement found in….*

- Means of escape (staircase and passageway)
- Escape aids provision (symbol)

4.7.1.3.4. Task C - Report Code Violations

Task Description:

After the compliance checking, the violations will be recorded in the building model for user to take necessary action to correct the design. This information can be presented graphically or in textual report.

Example Usage Scenario:

After the compliance checking, the violations will be recorded in the building model. The information can then be used subsequently by users to do the necessary corrections. Specific violation can be presented on screen as highlighted bounding box of the non-compliant building element with attached electronic “post-it-notes” to give detailed violation description. A summary report can also be generated in a written format to report all the violations found. The violation description will spell out the requirement to be met and what was actually found in the building model to the user. Prompt will also be given to the user to modify the design and rerun the compliance check.

4.8. ES-1 Cost Estimating

Processes Defined in this project:

1. Cost Estimating

Sub-Processes Defined in this project:

2. Scope Analysis
3. Identify Objects
4. Identify Tasks Needed to Install Objects
5. Identify Resources Needed to Perform Tasks
6. Quantify
7. Costing and Cost Summarization

4.8.1. Cost Estimating

4.8.1.1. Introduction

Overview:

These cost estimating usage scenarios describe the processes involved in determining costs based on information provided by objects in the Integrated Model. It includes:

- analysis of information in the model,
- adding information to help classify objects,
- adding objects to model tasks and resource usage,
- determining quantities,
- determining costs,
- and propagating costs and cost summaries back into the model.

Process Scope:

- The Cost Estimating section describes the major processes involved in producing an estimate. These include, Scope Analysis, Object Identification, Identification of needed Tasks and Resources, Quantification, Costing, and Summarizing Costs.

Out-of-Scope:

- The processes described here do not include cost attribute maintenance for actual costs incurred to a project.

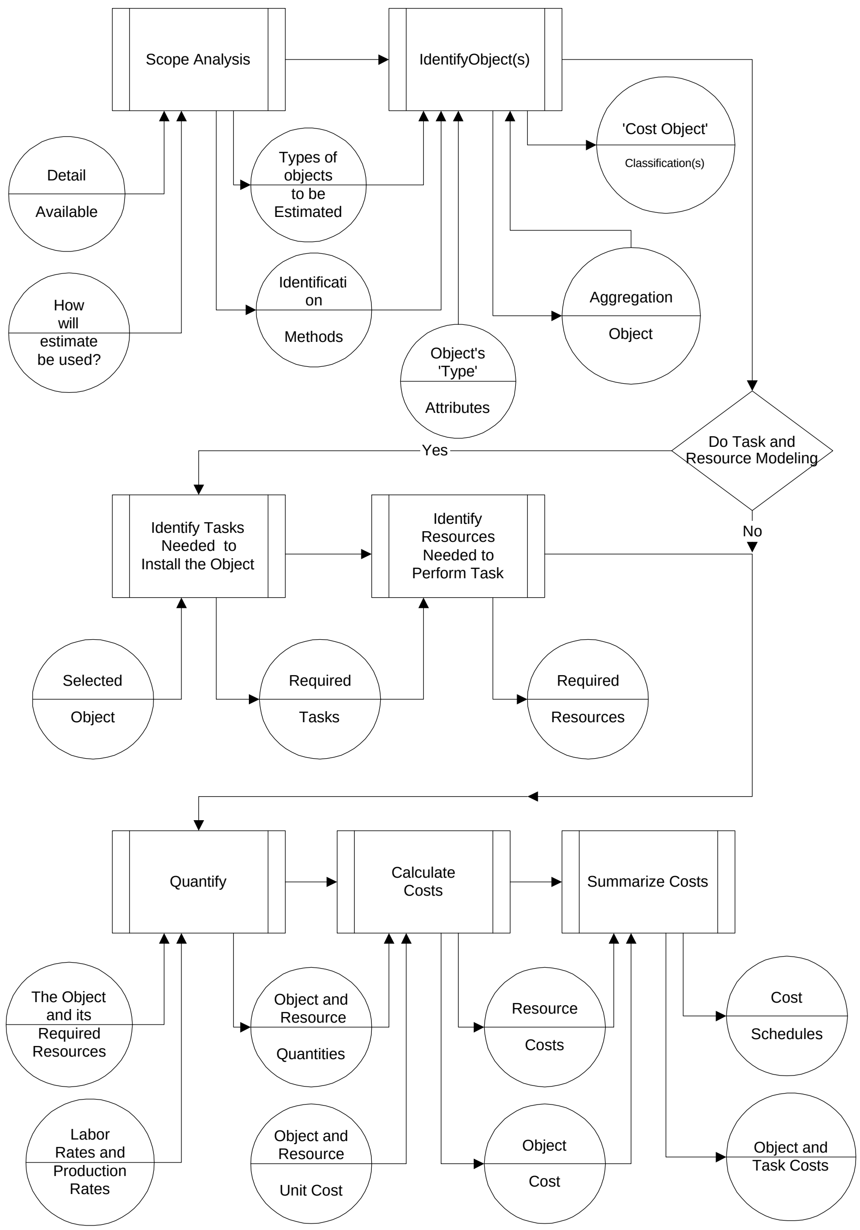
Definitions:

- None provided

Contributors:

- Mike Cole - NA
- Ray Brungard - NA
- Jeffrey Wix - UK
- Peggy Woodall - NA
- Annette Stumph - NA
- Roger Grant - NA

4.8.1.2. Process Diagram – Cost Estimating



4.8.1.3. Process Definition – Cost Estimating

4.8.1.3.1. Overview

Scope Analysis

This process looks at the level of detail that will be used in determining the objects that will be used in making the estimate. It provides information pertaining to the overall job, what state the design is in, and possibly for what the estimate is to be used for. This would indicate, for example, whether the estimate was to be detailed or conceptual.

Furthermore, it would indicate whether the entire project is being estimated, or just a portion of the project. It may be to estimate the cost impact of alternate designs, or a required change to the design.

Identify Objects

This process identifies the project objects to be costed. Such project objects may be an entire building, a section of a building, a space, individual elements (such as door), repeating types of elements (type of door), a process, or a resource.

Once the objects to be costed are selected, they must then be mapped to objects in the costing database. The mapping process may involve grouping model objects together (such as collecting connected wall segments to be estimated as a group). For construction planning purposes, construction zones are needed to define areas where similar objects need to be grouped for scheduling purposes. At this point you may need to replace a single object with an aggregation of its sub-elements (such as defining the separate concrete pours, and expansion joints of a foundation slab).

Identify Tasks Needed to Install Object

This process examines how the object (which was selected and identified in the previous step) is to built in the field and comes up with a set of tasks that are needed to install the object. For instance, a wall may require ‘framing’, ‘sheetrocking’, and ‘finishing’. These tasks can be modeled to come up with more accurate cost estimates. Information about the tasks may be used later in project scheduling.

Identify Resources Needed to Perform Task

Each task will require one or more resources. Resources may include labor (carpenters, electricians, …), equipment (crane, scaffolding, …), and materials (lumber, carpeting, …). Resource objects can be used along with task objects to model the costs of installing an object. Information about the resources may be used later in project scheduling.

Quantify

First, identify the way in which an object is to be ‘counted’, such as by piece, linear foot, etc. Next establish the amount of item to be measured. This includes the counting of discrete objects as well as calculation of quantities from the object’s dimensional information.

As with the object, its required resources also need to be quantified. Again one must establish the unit of measure for resources needed to install the and then establish the amounts of each resource that will needed to install the object. This will be calculated using the object’s dimensions and estimated resource usage based on those dimensions. Labor and material production rates will be used to establish the usage quantities.

Cost

This process evaluates the price impact of the objects. Using the quantities developed in the previous process, and applying unit costs for the overall object and/or its required resources, the cost impact of the object is calculated.

An object may have costs in addition to its cost per unit that has been calculated, such as tax, bond, insurance, and fees. Where appropriate, these costs should be added onto the cost of an object.

Summarize Costs

Report the estimated costs in a way that is easy for the intended customer to understand. This may include a schedule or report that organizes and summarize all of the cost information in the model, or possibly a browser that would allow a user to look at the cost information for any object in the model.

4.8.1.3.2. Task A - Scope Analysis

Task Description:

This process looks at the level of detail that will be used in determining the objects that will be used in making the estimate. It provides information pertaining to the overall job, what state the design is in, and possibly for what the estimate is to be used for. This would indicate, for example, whether the estimate was to be detailed or conceptual.

Once 'detail available' and 'estimate use' is analyzed, the types of objects to be estimated will be determined. For example, a 'conceptual estimate' may only look at project spaces and zones to come up with a rough initial estimate based on average cost per square meter. At a later stage of the design, more detailed estimating will be possible. The costs of individual walls, doors, windows, etc. can be modeled.

Some estimates are designed to determine the overall cost of a project. Others are used to do a value analysis among alternate designs or building methods. Late in the design process (and into the building process), estimates are needed to determine the cost impact of required design changes (change orders).

The classifications, attributes, dimensions, and context of model objects may be used to identify them and in turn map them to 'cost objects' in an estimating system. If the identification and classification system is to be highly automated, it should configured at this point to define how information about the objects will be used to map them to 'cost objects'.

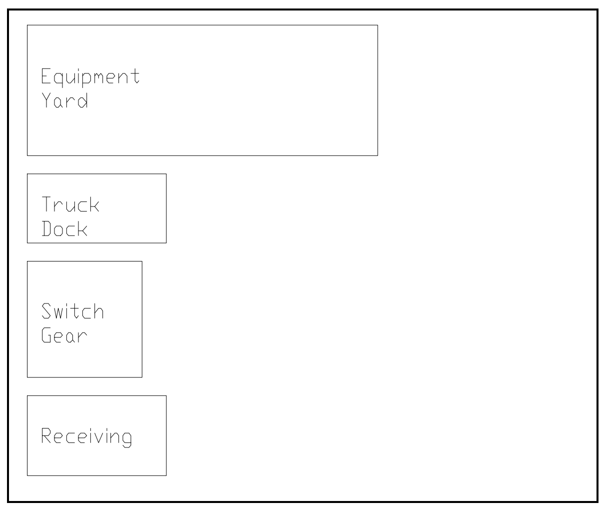
Example Usage Scenario:

Conceptual Estimate - Scope Analysis

At the beginning of the design process, an architect may only know the space and usage requirements of a building. In that case, a *conceptual estimate* would be done based on only this information.

If the model contains only space program information, it is determined that only the IfcSpaceProgramme objects will be used in this estimate.

In the diagram at the right you see some the objects that have been created to do a space planning analysis for the building.

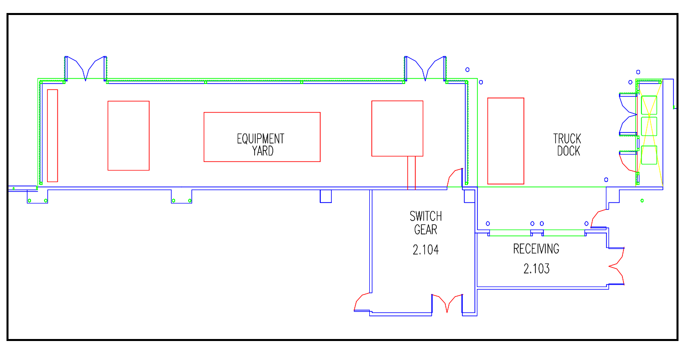


Example Usage Scenario:

Detail Estimate - Scope Analysis

As the design process progresses, actual construction objects (e.g. walls, doors, windows,…) are added to the model. We can use these objects to more accurately estimate the cost of constructing the spaces.

In the diagram at the right you see some the walls and doors that are now available for estimating. In a detail estimate these objects are targeted to be within the scope of the estimate.



At this time we determine whether the targeted objects will be estimated on a per unit basis, or if the tasks and resources to install an object will be modeled.

Example Usage Scenario:

Value Analysis - Scope Analysis

This frame will show a layout similar to the one use above, but with significant design changes to some part of it.

During the design process, the cost of alternate building designs must be evaluated. In this case, the objects that are removed, changed, or added, from one design to another must be targeted. If these objects are part of a grouping of construction objects that are installed as a unit, the entire grouping may be targeted.

4.8.1.3.3. Task B - Identify Object

Task Description:

This process identifies the project objects to be. Such project objects may be an entire building, a section of a building, a space, individual elements (such as door), repeating types of elements (type of door), a process, or a resource. The types of objects to be identified were determined in the 'Scope Analysis'.

Once the objects to be costed are selected, they must then be mapped to objects in the costing database. The mapping process may involve grouping model objects together (such as collecting connected wall segments to be estimated as a group). Conversely, it may involve delineating separate construction zones of a single object (such as defining the separate concrete pours of a foundation slab).

Using the object's type, attributes, dimensions, etc., determine what 'cost object' best models the costing of that object. The most flexible way to perform this task is to use an object browser and classify the objects manually. The quickest way is to define classification rules to automatically classify the objects. Each has advantages and disadvantages. The 'best' way may be some combination of these two.

Once the model object has been mapped to a 'cost object' classification, the classification should be recorded in the model object. This allows the identification process to be separated from the quantification and costing processes.

Example Usage Scenario:

Conceptual Estimate - Identify Objects

Model Object	**Estmating Object ID**
Equipment Yard	SP_EnclosedYard
Truck Dock	SP_Dock
Switch Gear	SP_Equip Room
Recieving	SP_Utility

Continuing form the previous example, the object identification process would examine the information in the IfcSpaceProgramme objects. Based on that information, it would select an estimating system object that would best model the cost of the space. The process stores the type of the estimating system object in the IfcSpaceProgramme object. Thus, later process do not need classify the object, they need only look up the estimating system classification stored on the IfcSpaceProgramme object.

Example Usage Scenario:

Detail Estimate - Identify Objects

Model Object	**Estimating ID**
Double Exterior Doors	DR_Ext
Double Interior Doors	DR_Int
Exterior Wall	WL_Ext
Interior Wall	WL_Int
Yard Wall	WL_Block

This step is the same for detail estimates. The model objects that have been targeted for estimating should be mapped to estimating system objects that are designed to represent them.

4.8.1.3.4. Task C - Identify Tasks Needed to Install the Object

Task Description:

Tasks are activities or operations required to place or install any object (permanent or temporary) in the project. To identify the tasks needed, the estimator selects a construction method for the object. The construction method will require one or more tasks to be performed. Task objects will be created and will be referenced the object to be constructed.

Example Usage Scenario:

Conceptual Estimate - Identify Tasks

This is beyond the scope of Conceptual Estimating, which generally relies on a cost per unit to estimate costs.

Example Usage Scenario:

Detail Estimate - Identify Tasks

Object	Required Tasks
Interior Wall	Framing
	Apply sheetrock
	Finishing
	Painting

Detail Estimates often model the tasks needed to install an object in order to estimate the cost of the object. If we are processing a wall object, several tasks may be identified and added to the model. These tasks may be; framing the wall, putting up sheetrock, finishing the sheetrock, and painting the wall. By modeling each of these tasks, we can later determine the cost of the wall. Modeling the tasks may be done outside the model, but by adding it to the model, we may be able to share this information with the scheduling process, which also models tasks.

4.8.1.3.5. Task D - Identify Resources Needed to Perform a Task

Task Description:

Each task will require one or more resources to be completed. Some resource types include Labor, Material, Subcontractors/Vendors, Equipment, etc. The quantity of the resource that is required and the unit cost of the resource, will contribute to the cost of the task.

The application will either create a resource objects, and/or select ones that already exist in the model. These resources will be referenced by the task to be performed. There may be multiple uses of a resource within the same task.

Example Usage Scenario:

Detail Estimate - Identify Resources

Task	Resources
Framing	Lumber
	Nails
	Carpenter

Continuing with the previous example, we now want to determine the resources required to complete each of the tasks. In the case of “framing the wall”, we may need various types of lumber, nails, and carpenters.

4.8.1.3.6. Task E - Quantify

Task Description:

The input to this process is the object that is to be quantified. Depending on the type of object, various dimension attributes will be used to calculate the overall quantity of the object and the quantities of resources required.

The ‘overall’ quantity of an object is measured in the dimension in which an estimator thinks of it in a ‘unit cost’ sense. For instance, the overall quantity of a wall might be in linear feet. The overall quantity of a concrete slab might be square feet or cubic yards, depending on how it is being estimated. The overall quantity should be calculated directly from the object’s dimension attributes.

The resource quantities are the amounts of the various resources needed to install the object. These quantities are based on the dimension and specification attributes of the object. For instance, a wall’s stud count will be based on the length of the wall and the stud spacing and possibly a waste factor stored in the estimating system. The duration quantities for the labor resources will be based on the object’s dimension

Model Object	Area
Equipment Yard	50 M2
Receiving	12 M2

Model Object	Length
Interior Wall	100 M
Yard Wall	30 M

Model Object	Resource	Quantity
Interior Wall	Framing Lumber	700 M
	Nails	1.5 Kilos
	Carpenter	20 Hours

When all of a task's resources are costed, the resource use costs should be accumulated and the task's overall cost should be updated.

The last cost to be calculated is the primary object's total cost. If no tasks or resources have been attached to the object to model its installation cost, and the object does not have 'parts' which have been costed, the object's cost will be based on its 'overall' dimension, various specifications, and a unit cost originating in the estimating system. The object's 'ProductCost' attribute should be updated with the calculated cost.

If the primary object's cost has been modeled using tasks, resources, or component 'parts' which contain costs, the object's 'ProductCost' should be updated to reflect these factors.

Example Usage Scenario:

Conceptual Estimate - Cost

In a conceptual estimate, you may compute cost directly from the quantity of an object. For example, if equipment yard space has a historical cost of $40 per M^2, you could estimate the cost based on the area, and that cost factor.

Model Object	Cost
Equipment Yard	$2,000
Receiving	$2,400

Example Usage Scenario:

Detail Estimate - Cost

If you are doing a more detailed estimate, in which you have done task and resource modeling, you would use the quantities of the various materials and resources and their unit costs to determine their costs. You would then total the costs of the materials and resources to determine the cost of the object.

Model Object	Resource	Cost
Intererior Wall	Framing Lumber	
	Nails	
	Carpenter	

4.8.1.3.8. Task G - Cost Summarization

Task Description:

When the process of determining costs is complete, we need to place the costs back into the model in a way that is understandable to people who need to review cost information. One place to put the cost is on the object whose cost was estimated. But this does not show the purpose or scope of the estimate, which is needed to understand the "meaning" of the cost. Placing estimated costs together in a "cost schedule" helps to give context and meaning to costs.

Once the costs are gathered together in a cost schedule, there should be a convenient way to reference the cost schedule elements from the object whose cost is being represented. Conversely, you should also be able to reference the object from the cost schedule element.

Example Usage Scenario:

Conceptual Estimate - Cost Summarization

In this example, the quantities and costs of the spaces programs that were targeted in the conceptual estimate are grouped and reported in a cost schedule. In this simple schedule, the cost of the spaces are listed at the detail level, and added together to show the cost of their grouping (Equipment/Receiving Area).

Estimate from Initial Space Plan			
Area	**Model Object**	**Object Quantity**	**Cost**
Equipment/Receiving Area			$10,900
	Equipment Yard	50 M2	$2,000
	Truck Dock	12 M2	$2,000
	Switch Gear	25 M2	$4,500
	Receiving	14 M2	$2,400

The model user should be able easily determine what objects the schedule refers to, for instance, an application may provide a utility to "jump to" a graphic representation of an object, from a display of the cost schedule. Conversely, from the graphic representation of an object, the user may wish to see a list of all cost schedules that contain information about the object's cost.

Example Usage Scenario:

Detail Estimate - Cost Summarization

In this example, the costs of the building elements that make up the spaces are targeted and reported together. The costs individual objects, of groupings (Equipment/Receiving Area) and sub-groupings (Walls and Doors) can all be reported in a single cost schedule.

The example shows one way that costs may be grouped but is in not intended as a suggestion for a specific hierarchy.

Again, it is important that a user be able to “flip” between a model object and the places where its cost is reported.

Estimate from Wall Layout Design

Area	**Type**	**Model Obect**	**Object Quantity**	**Cost**
Equipment/Receiving Area				
	Walls			
		Yard Wall		
		Exterior Walls		
		Interior Walls		
	Doors			
		Exterior Double Doors		
		Exterior Single Doors		
		Loading Bay Overhead Doors		
		Interior Single Doors		
		Interior Double Doors		
…	…	…		

4.9. [FM-3] Property Management *(Building Owner’s viewpoint)*

Processes Defined in this project:

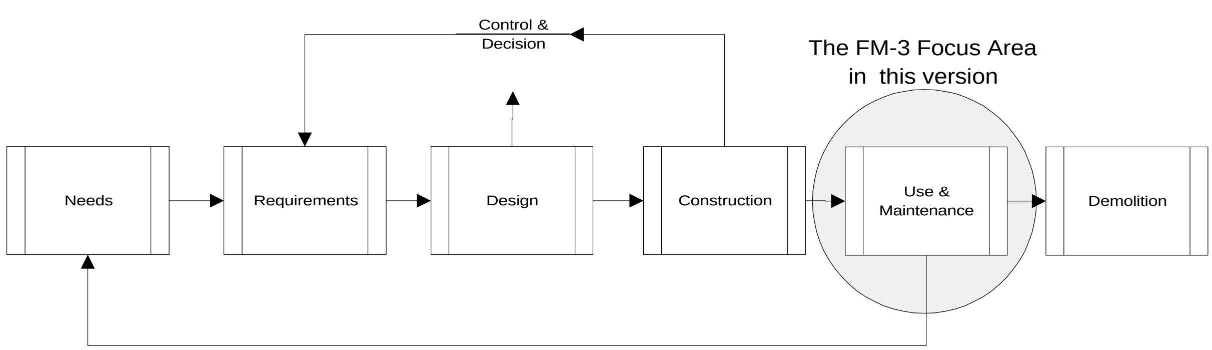
1. Enabling the use of IFC objects in property management
2. Grouping IFC objects
3. Linking the maintenance objects to the IFC objects

4.9.1. Property Management

4.9.1.1. Introduction

Overview:

Property management is a process starting from requirement programming and continuing through the building’s life cycle. The tools should facilitate the evaluation and comparison of properties and all costs during the construction and the life cycle. For these purposes the design and product data should be in such a format that it could be combined to the owner's and other external data bases for evaluation and management purposes.



Process Scope:

In this phase the FM-3 project covers just a subset of this process focusing on grouping of spaces and other possible objects for different purposes, like maintenance, administration, public registers, mapping etc. This process is based on objects provided by the design and construction process and uses mainly the attributes in the current model. The main user is the building owner and the benefits is more efficient use of the building data and through this cost savings in the administrative work. This process starts after the building is completed and is carried out through the whole life cycle of the building.

- Grouping IFC objects
- Linking the maintenance objects to the IFC objects

Out-of-Scope:

- Instructions for the maintenance
- Evaluation methods

Definitions:

- Group: a set of selected objects and / or groups
- Maintenance object: an object containing description, classification and maintenance history of linked IFC building elements

References:

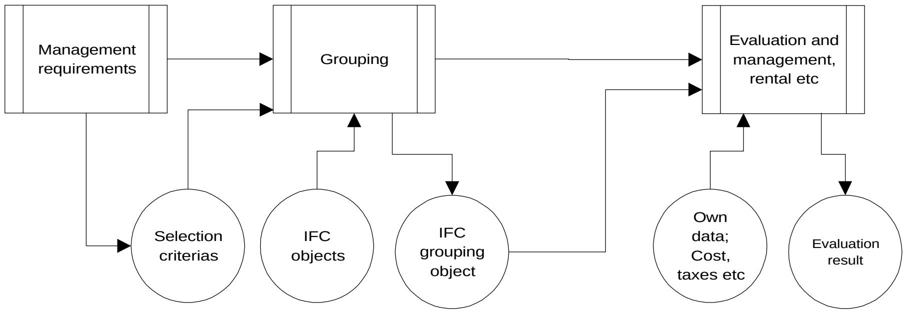
- Version 1.0 IFC Model
- CB-1 Client Briefing
- ES-1 Cost Estimating
- (FM-1 and FM-2 Maintenance, in version 3.0)
- FM-4 Occupancy Planning

Contributors:

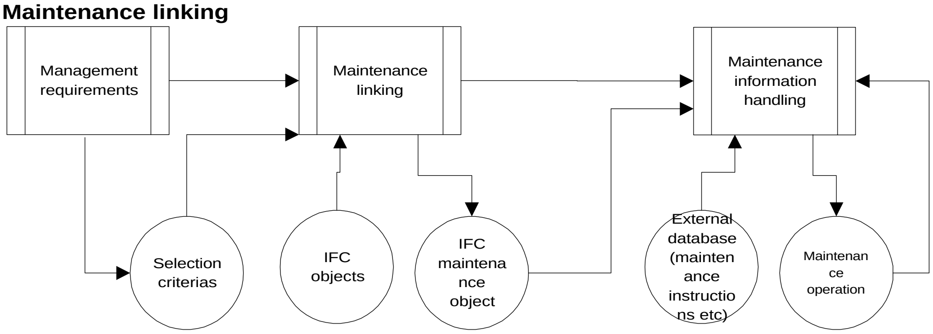
- Poul Sorgenfri Ottosen (AAU, Danish Department of Education) (Nordic)
- Jan Karlshoej (Carl Bro A/S) (Nordic)
- Arto Kiviniemi (VTT Building Technology) (Nordic)

4.9.1.2. Process Diagram – Property Management

Object grouping



Maintenance linking



4.9.1.3. Process Definition – Property Management

4.9.1.3.1. Overview

The need for grouping can be caused by any management purpose, like a new department, workgroup, cleaning area, renovation, fire zone etc. In this process the property manager can create new groups from selected objects. These groups can be used for any administrative or management purposes. All material or quantitative information is calculated from the IFC model. The model information can be used together with the owner's own or other external database information to evaluate operational costs or other needed values.

4.9.1.3.2. Grouping IFC objects

Task Description:

The first task is to define the grouping purpose, which defines the classification of this group. Then the objects for new groups can be selected through various methods:

- *any objects selected by the user*
- *filtered objects (type, properties or other selection key) selected by the user*
- *filtered objects in the whole model*

After the selection is completed the user can give a description to the group.

If the selected objects already belong to some group with the same classification, the application should warn the user about it and ask for instructions for further operations.

When the groups are formed the user can use those as the selection criteria for different operations and reports. All IFC object data should be available through these selections.

If the Grouping is added in IFC Release 1.5 this part may not be needed..

Example Usage Scenario:

The grouping mechanism enables many different functions in FM and also in other activities. With this mechanism the building owner can form for example rental, cleaning and other area combinations from spaces: The cleaning areas in the building needs to be defined. Different materials need different operations. The materials on surfaces can be recognized from the IFC model and picked automatically. After this the selected objects can be divided to proper sizes for operations and grouped to one unit. The classification and description of the unit enables easy administration, visualization and reporting of these units.

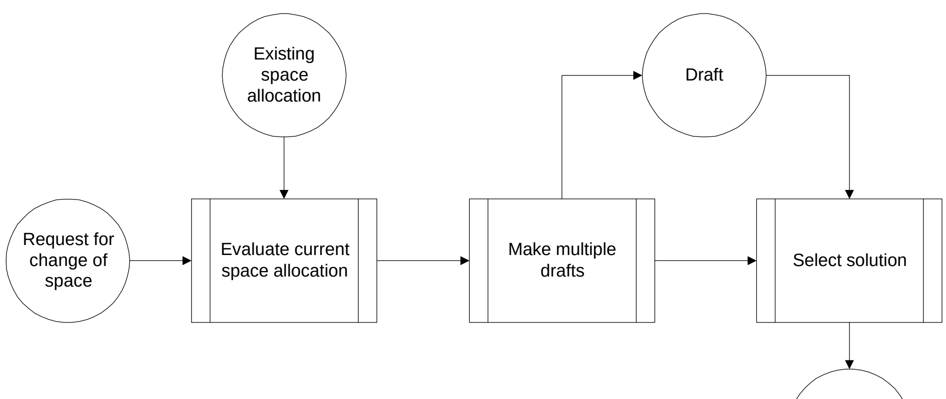
The building administrator is continually facing the problems of matching room size and facilities with user requirements. While the building will contain several types of rooms, each sized and equipped for a specific usage, each user will have individual requirements regarding for floorage and facilities.

The layout and facilities of a building are basically static. The building is the result of user or owner requirements at the time of erection. User requirements were analyzed and the building program, the design and construction of the building were based on the results. User requirements, however, do not remain static. They develop continuously according to the development of the individual user's business, which is independent of the physical setting provided by the building. The user can consequently be expected currently to increase or reduce his floorage requirements and/or require other facilities

Ideally, the building administrator should have a tool that can provide precise information about room types, users and historical development, including a facility for grouping together rooms used by the same institution or department. This information can be utilized by the building administrator for current inquiries and in creating a strategy for the further development of the building.

<u>Floorage control tool - examples of usage</u>

The administrator will continually evaluate the degree to which the building is utilized, for example by dividing the rooms into categories, which would enable him to calculate the net/gross area ratio. This can be used both in the operation of the existing building, as a guide for refurbishment and as an element of the planning of new buildings.



The possibility of carrying out area analyses helps to improve the planning of extensions to buildings, the purchase of additional buildings and the erection of new buildings. The area analyses are also useful for the building administrator in deciding whether a change in the distribution of existing rooms among users would benefit

University Group	Member
Department 1	Room 1 Room 2 Room 4
Department 2	Room 3 Room 5 Room 6
Department 3	Room 7 Room 8
Institute 1	Department 1 Department 2 Department 3

himself and/or the users or increase the possible uses of the building. See also the process diagram below.

Besides a room index based on size and type, the building administrator will want to make indexes grouping room types by functions. If the group is recorded in the database or in a model instead of just being stored as an inquiry, it is possible for independent systems to use the information

A size index can be used directly as a basis for planning a cleaning schedule, including outsourcing/overall planning/management and the practical aspects of daily cleaning.

Residence Group	Member
Apartment 1	Room 101 Room 102 Room 103
Apartment 2	Room 201 Room 202 Room 203
Apartment 3	Room 301 Room 302 Room 303
Section A	Apartment 1 Apartment 2 Apartment 3 Room 001 (Corridor)

University Group	Member
Laboratory	Room 801 (Electrical) Room 802 (Structural) Room 903 (Chemistry)
Dark-room	Room 201 Room 202 Room 203
Special rooms	Laboratory Dark-room

4.9.1.3.3. Linking the maintenance objects to the IFC objects

Task Description:

First task is to define the selection criteria for a maintenance group. Then the objects for a new group can be selected through various methods:

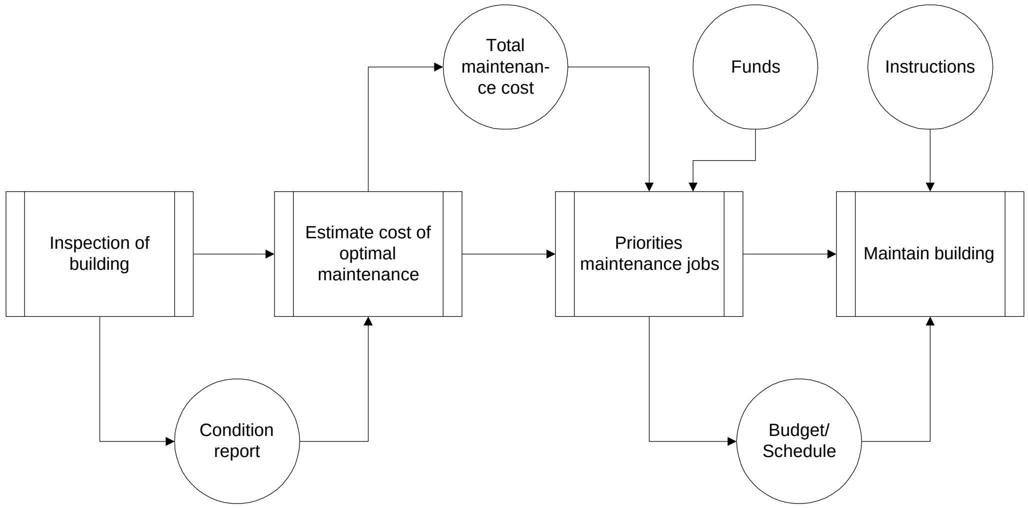
- *any objects selected by the user*
- *filtered objects (type, properties or other selection key) selected by the user*
- *filtered objects in the whole model*

After the selection is completed the user can give a description to the new maintenance group.

If the selected objects already belong to some maintenance group, the application should warn the user about it and ask for instructions for further operations.

When the maintenance groups are formed the user can use those as the selection criteria for different maintenance operations and reports. All maintenance data is stored in the maintenance object and the IFC object data should be available from the actual objects.

Example Usage Scenario:



The different window and door types as well as other objects in the building can be linked to a maintenance object. The grouping mechanism is identical to the grouping activities.

With the description and classification attributes the different maintenance groups, purposes and needs can be identified. With delivery date and guarantee ending date the status of the guarantee can be stated. The guarantee terms can define the maintenance period and the maintenance instructions. With last maintenance date, maintenance handling and maintenance history the operations can be verified. The inspection intervals, last inspection date, inspection handling and inspection history helps the property manager to plan and define priorities for maintenance operations in connection with the information in cost object.

The guarantee terms, maintenance instructions, maintenance history and inspection history are pointers which refer to external databases or paper documents.

The dates of delivery, guarantee ending, last maintenance and last inspection as well as maintenance and inspection handling enable the search, selection of objects and operation to them on these criterion.

To plan and carry out facility maintenance a knowledge of the condition of the building and the funds available for the purpose is necessary. The Building administrator will base his decisions regarding building maintenance on inspections, which together with the funds/appropriations available form the basis of the budget and, consequently, the maintenance schedule for the budget period.

A typical maintenance sequence is shown below. The buildings are inspected periodically to record its condition and maintenance requirements, including whether the maintenance measures in question are preventive, remedial or reconstruct. The inspection results in a condition report, which the building administrator can use for a total price calculation. The calculation is compared with the funds available, which will enable the building administrator to prioritize maintenance jobs if necessary, in order not to exceed the total budget. The building administrator will have to evaluate the costs that will result from omitting a job, such as increased future maintenance costs and the possible expire of periods of guarantee. The result of this process is a budget which will also be a maintenance schedule. The physical maintenance can be implemented according to instructions provided either by the producer of the component in question or the instruction of the building administrator.

Condition report					
SfB-Classification group	*Material*	*Component*	*Supplementary description*	*State*	*What to do*
21(Facade)	Brick			OK	
22(Internal wall)	Brick		Crack above tile	Need repair	putty and painting
31(Window)	Steel		Rust in the bottom of the window frame	Need repair	cleaning, painting

4.10. [FM-4] Occupancy Planning

Processes Defined in this project:

1. Occupancy Planning
2. Design of Workstations
3. Floor Layout of Workstations for an Open Office

4.10.1. Occupancy Planning

4.10.1.1. Introduction

Overview:

The occupancy planner (includes interior designers, facilities managers, architects, furniture dealers' designers, etc.) applies standards during the assignment of people and organizations to interior spaces. It also involves the planning and moving of building assets such as equipment and furniture. This process occurs during the initial planning of space occupancy, and whenever that occupancy needs to change (company reorganization, company growth, or new hires, etc.). The layout and design of typical workstations can be sub-processes of the occupancy planning when it involves systems furniture planning for open offices. These processes require information about the building floor spaces. They also generate space occupancy data for future use of office planning.

Automatic input and utilization of the IFC supported object data, such as building elements and spaces as well as FF&E and occupants, may improve the efficiency of the processes. New objects generated will also be IFC compliant so that they can be used by varies FM processes during the operation of the facility.

Process Scope:

- Evaluation of open spaces
- Move of people and FF&E

Out-of-Scope:

- Design of workstations
- Floor layout of workstations
- Stacking and blocking
- Work and purchase order tracking process

Definitions:

- FF&E: furniture, fixture, and equipment that is movable.
- Schematic Design: the conceptual allocation of space to define adjacencies and required functions defined by area and circulation paths.
- Move Plan: a plan that is used in Facilities Management for occupancy planning, moving people and FF&E around.

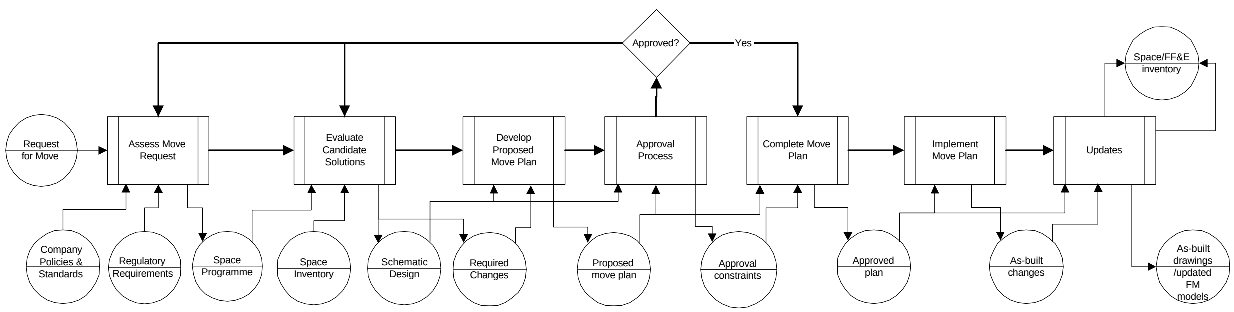
- Candidate Solution: existing or new spaces, as well as all the changes, work required, FF&E to be moved or purchased, occupants to be moved, that meet the requirements defined in the space programme.

References:

Contributors:

- Sandy Anderson (IBM) (NA) (temporary)
- Rick Bartling (Herman Miller) (NA)
- Vicky Borchers (MKS) (NA)
- Francois Grobler (USA-CERL) (NA)
- Jeff Laba (IBM) (NA) (temporary)
- Paul Lewis (Visio) (NA) (temporary)
- Elizabeth Menard (Naoki Systems) (NA)
- Richard See (Autodesk) (NA) (part-time)
- Rob Wakeling (Visio) (NA)
- Kevin Yu (Naoki Systems) (NA)

4.10.1.2. Process Diagram



4.10.1.3. Process Definition

4.10.1.3.1. Overview

The occupancy planner (includes interior designers, facilities managers, architects, furniture dealers, etc.) applies standards during the assignment of people and organizations to interior spaces. This process occurs during the initial planning of space occupancy, and whenever that occupancy needs to change (company reorganization, company growth, etc.).

4.10.1.3.2. Task A - Assess Move Request

Task Definition:

Assess request with respect to occupant information, company policies, and regulatory requirements. Identify FF&E required for the occupant, and generate space programme.

Example Usage Scenario:

The first step is to assess the move request. In this step, the occupancy planner evaluates the request with respect to occupant information, input for impact assessment for future move consolidation, company policies, and regulatory requirements. This step may identify the FF&E required, and finally generate the space program for the request.

4.10.1.3.3. Task B - Evaluate Candidate Solutions

Task Definition:

Compare space programme to available (incl. existing or added) spaces to find candidate solutions including the changes of spaces and FF&E, required work, occupants to be moved, etc..

Example Usage Scenario:

The second step is to evaluate candidate solutions based on criteria such as optimal space use, adjacency and proximity, and future BPR (Business Process for Re-engineering) changes. The space program from the last step is used to block plan available spaces, and find candidate solutions that include the changes of spaces and FF&E. This process will also result in schematic designs.

4.10.1.3.4. Task C - Develop Proposed Move Plan

Task Definition:

During the design and generation of drawings, we allow for client review and approval. Define temporary staging areas, generate schedules, identify sources of all FF&E required and generate a cost estimate.

Example Usage Scenario:

An occupancy move plan should be developed in this step to allow for client review and approval. A list of all FF&E required is created. A preliminary work schedule and a cost estimate will be included in the plan. The schematic design used in the last step will also be included in the plan package.

4.10.1.3.5. Task D - Approval Process

Task Definition:

Occupant and management review proposed move plan and either approve (possibly with constraints) or rejects --> revert to previous steps.

Example Usage Scenario:

The approval process involves the review of proposed plan. This process could either approve (possibly with constraints) or rejects. In the case of rejection, it is possible that the move request is re-analyzed or the candidate solutions are re-evaluated.

4.10.1.3.6. Task E - Complete Move Plan

Task Definition:

Modify proposed plan to comply with constraints. Generate work orders and purchase orders.

Example Usage Scenario:

If the plan has been approved, there is a need to modify the proposal as with the constraints suggested. The work orders and purchase orders will be generated, and a new plan will be developed. Based on the new plan, bills-of-materials for the purchase of new FF&E will be produced.

4.10.1.3.7. Task F - Implement Move Plan

Task Definition:

Purchase FF&E. Perform work orders. Deal with change orders. Complete staging space. Move the occupant.

Example Usage Scenario:

The space occupants including the existing FF&E will be moved. During the implementation, as-built changes will be summarized and possibly updated into the original move plan. The implementation will eventually result in new or revised space and FF&E inventories.

4.10.1.3.8. Task G - Updates

Task Definition:

Revised documentation and databases to reflect new and revised spaces and assets.

Example Usage Scenario:

Finally, documents and databases of space and asset (i.e. FF&E) inventory will be updated to reflect the changes.

4.10.2. Design of Workstations

4.10.2.1. Introduction

Overview:

The facility manager (also interior designers, architects, furniture manufactures and designers, contract furnishing dealer, etc.) designs typical workstations to be used by office staff. The workstations designed could be used as company standards and be selected in the layout of the systems furniture. This process could also occur in the entire process of occupancy planning in an organization.

Process Scope:

- Approval of design
- Systems furniture design

Out-of-Scope:

- Design of workstation groups
- Layout of workstations
- Stacking and blocking
- Standardizing workstations
- Technology configuration

Definitions:

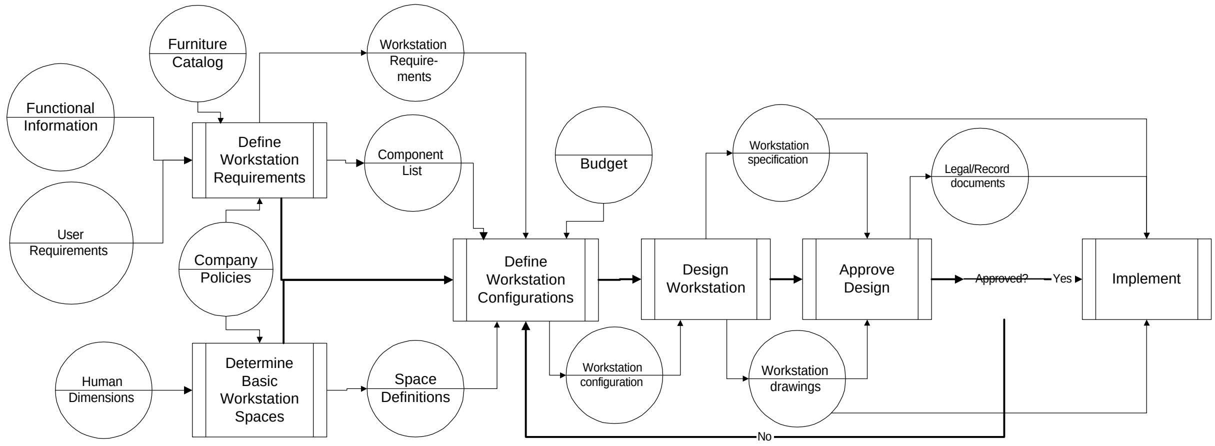
- FF&E: furniture, fixture, and equipment that is movable.
- Systems furniture: is meant to represent furniture systems that integrate both modular and free-standing furniture with independent panels hanging furniture components such as work surfaces, storage, and so on.
- Workstation: a bound space assembled by systems furniture or floor to ceiling wall systems with necessary office equipment to house one or a few people to perform tasks.

References:

Contributors:

- Sandy Anderson (IBM) (NA) (temporary)
- Rick Bartling (Herman Miller) (NA)
- Vicky Borchers (MKS) (NA)
- Francois Grobler (USA-CERL) (NA)
- Rob Wakeling (Visio) (NA)
- Kevin Yu (Naoki Systems) (NA)

4.10.2.2. Process Diagram



4.10.2.3. Process Definition

4.10.2.3.1. Overview

The facility manager (also interior designers, architects, furniture manufactures and designers, contract furniture dealers, etc.) designs typical workstations to be used by office staff. The process starts from defining the functional requirements of the workstation based on the work types of the employees who use the workstation. The workstation to be designed must also meet the requirements of basic human dimensions for spaces. Special requirements such as a wheelchair must be considered. The design drawings and specifications should be produced based on the configurations of the workstation components and equipment. Final design must be approved before implementation.

4.10.2.3.2. Task A - Define Workstation Requirements

Task Description:

Define the basic component and equipment types, the security, privacy and special requirements according to the employee type, work types, and company policies, etc..

Example Usage Scenario:

The first step is to define the functional requirements of the workstation. This requires information about type of the user (e.g. a computer programmer), and the type of work (e.g. design and programming) he or she performs. A functional information worksheet (see Table 1) can be used for collecting the information.

Ergonomic requirements for particular types of users will also be considered such as that a wheelchair must be used, or that the height of individual worksurface is specially required. Some companies may also want to apply some special company policies for this process; an example is the style of furniture for managers, etc..

Table 1: Sample Functional Information Worksheet

Employee Name:	Jack Smith		
Employee Type:	Computer Programmer		
Work Task Description:	Analysis	Programming	Internet access
Storage Items:	Books References Accessories	References	
Components Required:	computer surface write/read surface storage	computer surface	modem
Equipment Required:	PC computer	PC computer	
Average Weekly Hours:	10	20	2.5
Special requirements:	No		

Based on the information provided, a list of basic furniture components will be generated such as types of work surfaces, file storage, panel partitions, lighting and seating. In addition, a list of office equipment types will also be created. For example, a programmer will need a computer; and based on the work types, the computer may need a modem.

The workstation requirements will be summarized that include security requirements (e.g. files must be locked), electrical and telecommunication requirements (e.g. 3-circuit, dedicated, network type, etc.), privacy requirements (e.g. visual privacy), and any types of special requirements such as aesthetic requirements. Table 2 shows a sample list of workstation component and equipment types for a computer programmer.

Table 2: List of Component and Equipment Types

Furniture Component:	
worksurface	writing & reading PC computer file references accessories
Storage	filing storage reference storage
Seating	main chair (1)
Panel type	moderate privacy enclosure
Lighting	ceiling lighting
Equipment:	
PC Computer	1 (with modem)
Special Requirements:	
Worksurfac e	Worksurface height is 38 inches as requested by Jack

4.10.2.3.3. Task B - Determine Basic Workstation Spaces

Task Definition:

Define spaces of the workstation (including circulation space inside of the workstation) according to the basic requirement of human dimension standards, and company policies.

Example Usage Scenario:

The basic spaces according to the human dimension standards requirements will be determined in this step. This step can be performed in parallel with the first one. Table 3 shows some examples of human dimension requirements for a basic workstation. One may also want to apply some company policies to this step.

Table 3: Human Dimension for Basic Workstation

Zone/Height	Dimension (inch)
Worktask Zone	66 - 70
Chair Clearance Zone	66 - 70
Circulation Zone	24
Worksurface Height	29 - 35
Shelf Height	?

4.10.2.3.4. Task C - Define Workstation Configurations

Task Definition:

Finalize all workstation components with all detailed dimensions and material information, and spaces.

Example Usage Scenario:

After the above two steps have been finished, detailed workstation configurations will be designed, which include all the information about the components (see Table 4), equipment, and spaces (i.e. their dimensions, materials, space footage, and even brands, suppliers, models, colors, etc.), as well as acoustical and electrical properties.

Table 4: Sample Component Configuration of Workstation (unit: inch)

Item	Dimension			Hanging/Mounting	Finish/		
Description	Height	Width	Depth	Height	Color	Quantity	Remark
Overhead storage with task light	18	42	15	55	walnut	2	
Worksurface rectangle	N/A	42	42	38 (on request)	walnut	2	Special request
Worksurface square corner	N/A	42	48	38 (on request)	walnut	1	Special request
Chair	adjustable	22	22	N/A	blue f[illegible]	[illegible]	[illegible]
Panel	76	42	2.5	N/A	grey f[illegible]	[illegible]	[illegible]

4.10.2.3.5. Task D - Design Workstation

Task Description:

Produce the workstation drawings and define the specifications according to the configurations.

Example Usage Scenario:

Based on the configurations defined in the last step, design drawings and specifications will be produced in this step. See Figure 1 for an example of a workstation drawing.

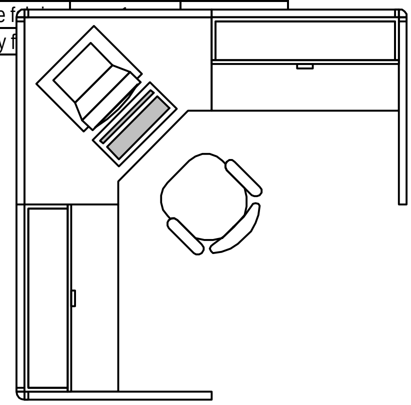


Figure 4: Sample Workstation

4.10.2.3.6. Task E - Approve Design

Task Description:

Corporation and also the user approve the design.

Example Usage Scenario:

This process examines the design produced in the last step and attempts to approve it or reject it based on criteria.

4.10.2.3.7. Task F - Implement

Task Description:

Implement the design.

Example Usage Scenario:

After the design has been approved, the implementation will be executed.

4.10.3. Floor Layout of Workstations for an Open Office

4.10.3.1. Introduction

Overview:

The facility manager (also interior designers, architects, or furniture dealers, etc.) designs the layout of the workstations for an open office. The purpose of the design is to organize the individual workstations into workstation groups with each usually representing a departmental unit and performing a certain type of function as a whole such as marketing. The workstation groups are assembled workstations connected by shared vertical panels. In order to group the workstations, employee work interaction patterns must be captured and block plan mechanism may be used. The process is part of the entire floor furniture and equipment planning for the department(s), and occurs after typical individual workstations have been designed.

Process Scope:

- Floor blocking

Out-of-Scope:

- Bubble diagram design
- Design of workstations
- Standardizing of workstations
- Stacking
- Approval process of the design

Definitions:

- FF&E: furniture, fixture, and equipment that is movable.
- Workstation: is a bound space assembled by systems furniture or floor to ceiling wall systems with necessary office equipment to house one or a few people to perform tasks.
- Systems furniture: is meant to represent furniture systems that integrate both modular and free-standing furniture with modular contiguously grouped panels hanging furniture components such as work surfaces, storage, and etc..
- Workstation group: physically adjacent workstations that together perform a certain function, such as marketing, or computer programming.
- Floor blocking: the process of designing block plans that are two-dimension horizontal layout in a floor plan. Each block is a large bound zone that contains one or more work groups that represent a functional unit such as a department, etc..
- Bubble Diagram: a graphical diagram that depicts the adjacency relationships of work groups (e.g. departments) using circles and lines without floor space restrictions. The circles (i.e. bubbles) are used to represent the work groups by their size, color, or shading meaning the size and name of the group. The lines are used to represent the closeness of the adjacency usually by their thickness.

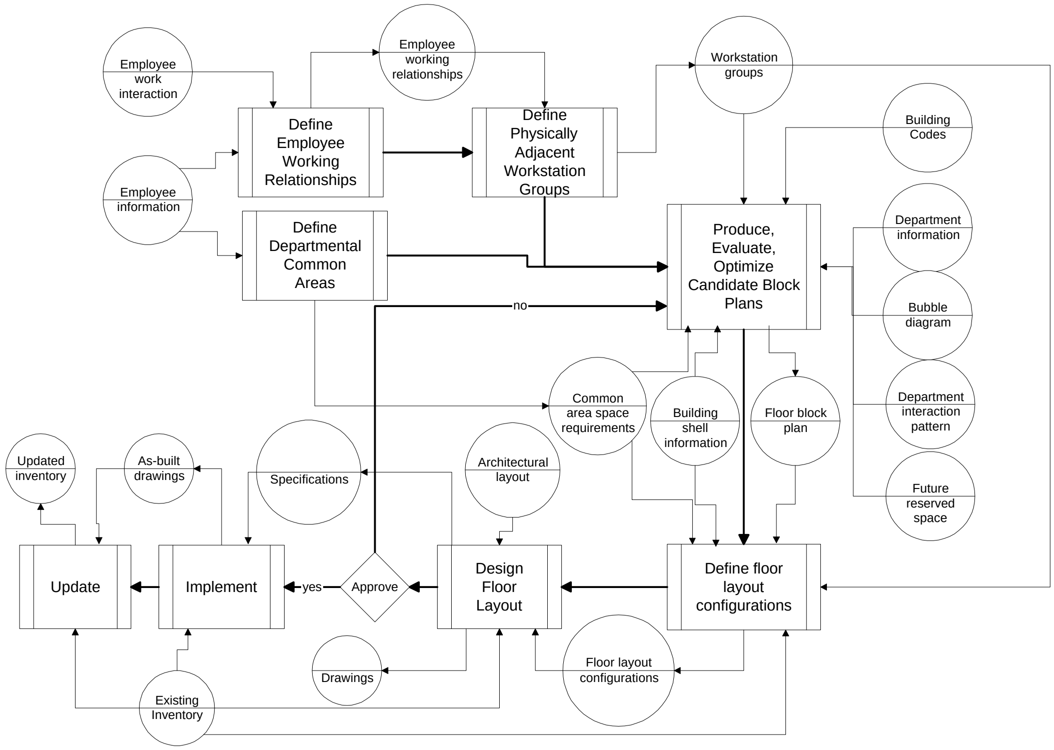
References:

- Julie K. Rayfield, “The Office Interior Design Guide – An Introduction For Facilities Management &

Contributors:

- Sandy Anderson (IBM) (NA) (temporary)
- Rick Bartling (Herman Miller) (NA)
- Vicky Borchers (MKS) (NA)
- Francois Grobler (USA-CERL) (NA)
- Rob Wakeling (Visio) (NA)
- Kevin Yu (Naoki Systems) (NA)

4.10.3.2. Process Diagram



4.10.3.3. Process Definition

4.10.3.3.1. Overview

The facility manager (also interior designers, architects, or furniture dealers, etc.) designs the layout of the workstations for an open office. The process starts from defining the employee working relationships so that closely related workstations can be adjacently assembled into workstation groups. Common departmental areas such as circulation or service areas must be considered. The adjacency relationships between the departments or workstation groups must be determined. It is usually necessary and efficient to use block plan mechanism to mark the floor area into different and big plane blocks with each representing a departmental unit, such as a research department. Workstations and groups will then be fit into certain blocks. Actual design drawings and specifications of the workstation layout will be produced based on the workstation layout configurations. The design must be approved before implementation.

4.10.3.3.2. Task A - Define Employee Working Relationships

Task Definition:

Define the individual employees working interaction patterns and meeting frequencies according to the work they perform.

Example Usage Scenario:

Table 5: Sample Worksheet of Employee Interaction

Employee Information			Interaction	With whom			Importance	Average	Daily
Name	Dept.	Position	Description	Name	Dept.	Where	Rating	Dur.	Freq.
Jack	Dev.	System Designer	Program Corporation	Tony	Development	either office	3	30 min.	0.5
			Consulting	Kevin	Research	Kevin	2	30 min.	0.25
			Customer Requirement	Linda	Marketing	either	1	5 min.	1
			Approval	Jeff	Project Manager	Jeff	4	20 min.	0.25
…		…	…	…	…		…	…	…

An employee working interaction pattern summary is produced in this step. This summary includes information such as department name of the employees, with whom one has interaction with the other, how many times of such interaction, and average duration of each meeting. A worksheet (see Table 5) can be used for collecting information and interaction analysis.

From this information, an adjacency diagram may be used to clarify relationships and provide a basic schematic for further development of a floor plan (see Figure 2).

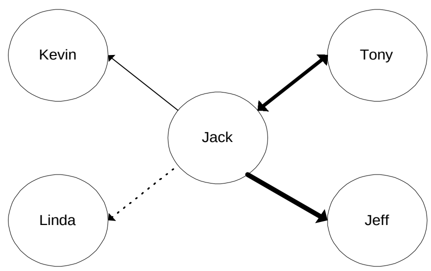


Figure 5: Adjacency Diagram

4.10.3.3.3. Task B - Define Physically Adjacent Workstation Groups

Task Definition:

Define the functional workstation groups according to the individual employees working relationships summarized in the last step. A group consists of one or a few different types of typical workstations that have close working relationships, frequent or infrequent but critical interactions, and perform the same kind of functions. In the case that the interaction frequency and importance rating are conflict, a decision has to be made based on human judgement.

Example Usage Scenario:

Once the employee working interactions have been determined, the physically adjacent workstation groups can be defined with each providing a certain working function (e.g. development group). Each workstation group consists of one or a few different types of typical workstations that have close working relationships, frequent interactions, and perform the same kind of function as a whole. Adjacent workstation groups are typically connected by shared vertical panels and necessary connecting accessories. See Figure 3 for an example (*Note: this diagram doesn't show the actual results of the interaction relationships from the last step. Real world project examples may be provided later.*)

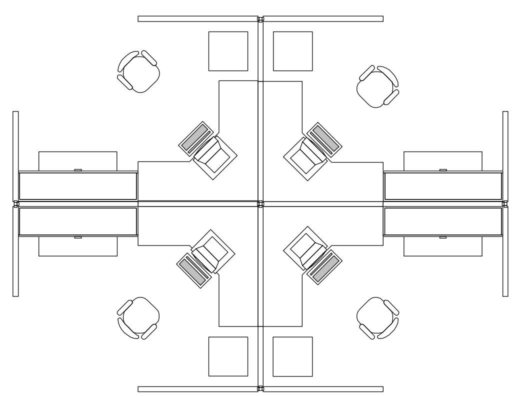


Figure 6: Sample Workstation Group

4.10.3.3.4. Task C - Define Departmental Common Areas

Task Definition:

Define the areas that are shared by all employees in the department, such as common circulation and conference rooms, etc.

Example Usage Scenario:

The departmental common area such as the common circulation areas, conference rooms, printing service center, etc. is defined. This step can be performed independently with the first one. The common area requirements will be used for floor layout design in later steps.

Table 6: Sample Adjacency Matrix

Accounting	3	2	2	1	1
	Marketing	3	2	2	1
		Executive	3	2	2
Adjacency Rating Code:			**Operation**	2	2
1. Undesirable				**Research**	3
2. Desirable					**Development**
3. Essential					

4.10.3.3.5. Task D - Produce, Evaluate and Optimize Candidate Block Plans

Task Definition:

Segment large spaces for workstation groups according to the relationships between the workstation groups, and the relationships between departments in case of multiple departments. Floor geometry constraints such as column grids, ceiling grids, window grids, the space footage must be taken into consideration. A floor block can contain one or more workstation groups, or one or more workstations. This step also evaluates different candidate block plans and attempts to optimize the space or FF&E usage.

Example Usage Scenario:

After the above three steps, a floor block plan can be designed according to the relationships between the workstation groups, and the relationships between departments in case of multiple departments. The relationships between the workstations can be determined through the 'Adjacency Matrix' mechanism (see Table 6). Different adjacency rating scheme can be used. Building shell information such as column grids, ceiling grids, window grids, the space footage should be essential as input information for this design. Different candidate plans will be evaluated according to criteria of optimum use of space or FF&E. Eventually, each block could contain one or more workstation groups as well as individual workstations that do not belong to any defined group. Figure 4 is a sample that shows a portion of a floor block diagram.

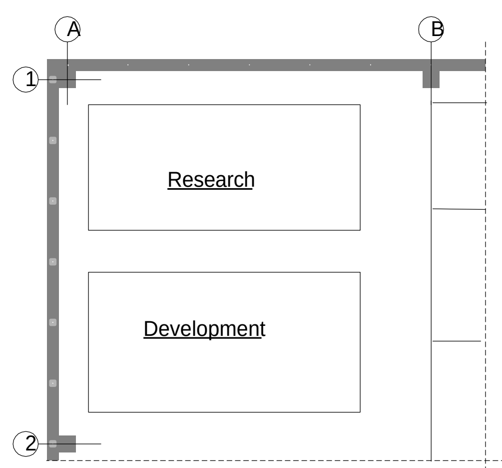


Figure 7: Sample Floor Block Diagram

4.10.3.3.6. Task E - Define Floor Layout Configurations

Task Definition:

Define all the detailed footage of all the workstations, workstation groups and departmental boundaries.

Example Usage Scenario:

In order to do the layout design, detailed floor layout configurations must be defined, which includes all the detailed footage of all the workstations, workstation groups, and departmental boundaries on the floor. The furniture system chosen during the floor layout configuration will affect the workstation boundaries.

4.10.3.3.7. Task F - Design Floor Layout

Task Definition:

Produce the workstation layout drawings and define the specifications.

Example Usage Scenario:

Based on the configuration, the final step is to perform the design of floor workstation layout. From this step, drawings and specifications will be produced. Figure 5 shows a portion of a floor layout of workstations as an example.

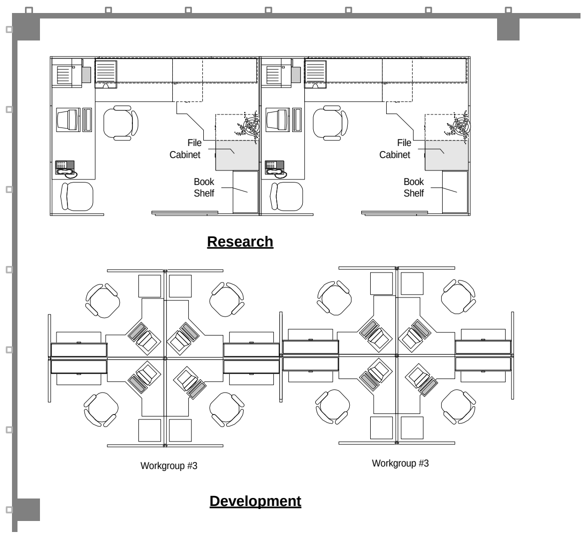


Figure 8: Sample Floor Layout of Workstations

4.10.3.3.8. Task G - Implement

Task Definition:

Implement the design of workstation layout.

Example Usage Scenario:

Once the design is approved, it is implemented based on the design drawings and specifications. During the implementation, existing inventories should be considered. Drawings are changed and as-built drawings are produced throughout the implementation.

4.10.3.3.9. Task H - Update

Task Definition:

This is an on-going process that occurs during the course of design and implementation. Inventories are updated.

Example Usage Scenario:

As-built drawings are updated; any existing inventories are updated with left over furniture from the job.

4.11. [SI-1] Photo Accurate Visualization

Project process list:

- Photo Accurate Visualization

4.11.1. Photo Accurate Visualization

4.11.1.1. Introduction

Overview: In the design of a building or other structure, the architect or designer may want to see what the building or the structure will look like, or may want to render images for the client's benefit. Such visualization may be desired at any time from the earliest architectural design or retrofitting to the final interior design. Visualization is the key to solving lighting and daylighting design problems, and is also important in assessing building performance and human comfort issues.

Process Scope:

- Selection of surface materials
- Selection of lighting
- Rendering

Out-of-Scope:

- Process of acquisition of space/building geometry
- Photometric information that may be generated by the application used in the simulation

Definitions:

- None provided

References:

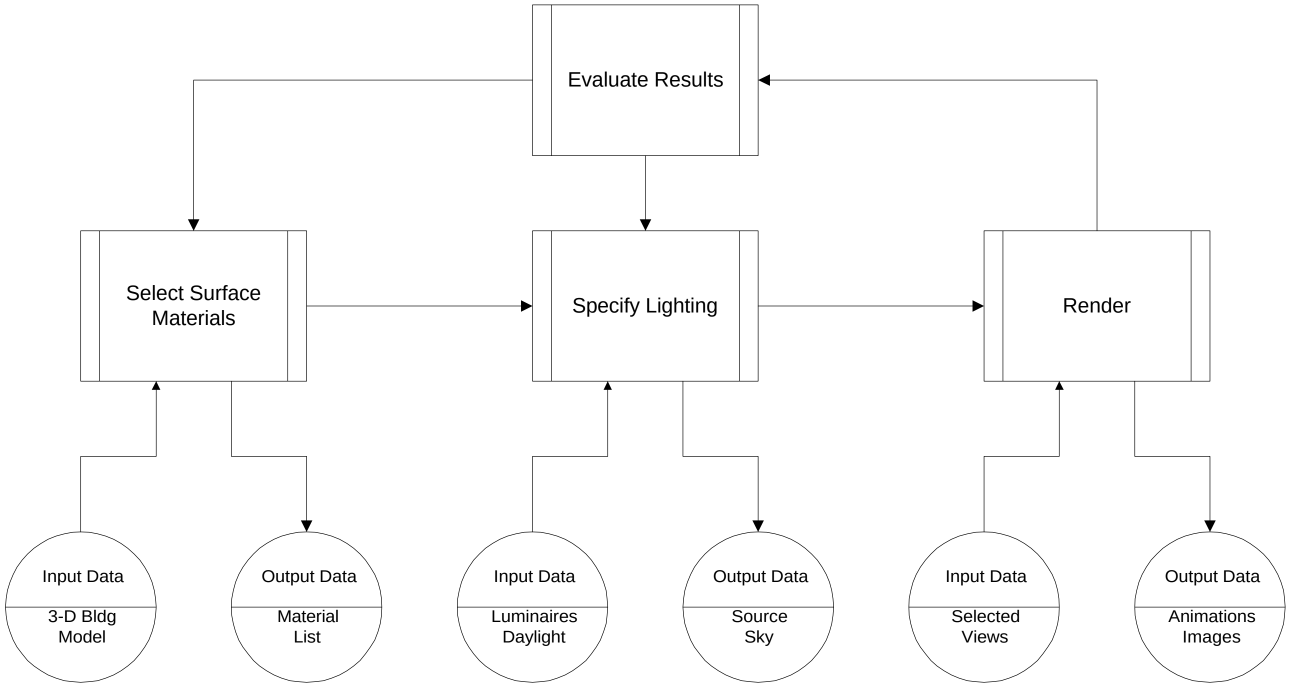
- None provided

Contributors:

- Vladimir Bazjanac (LBNL) (NA)
- Greg Ward (LBNL) (NA)

Note: Contributors listed here are people who have contributed to the definition of this process to-date; more names may be added later.

4.11.1.2. Process Diagram



4.11.1.3. Process Definition

Overview

The input information for the selection of materials consist of the three-dimensional representation of space or building geometry. Each of the surfaces that affect the rendering is associated with a particular material, for which reflectance, transmittance, color, pattern and texture are defined.

To perform a visual simulation, the user selects and places light sources (luminaires) in three-dimensional space, and specifies daylight conditions. Light source configuration and light distribution data are selected from manufacturers' catalogs. The sun and sky conditions (sky distribution and solar position specific to time in the simulation) are taken from a set of quantitative models (including daylight models) appropriate to the building site.

To define the rendering, the user also specifies a point in three-dimensional space from which the space or the building are viewed. The user may also specify the animation path, should he wish to create an animation. The output from the simulation are two-dimensional (floating point) color images, luminance and isolux contour plots, and/or animations.

The input of three-dimensional geometry description of the space or the building, if done manually, is very time-consuming an error prone. So is, to a lesser extent, the manual input of material and surface properties. If these data are originally input into IFC-compliant CAD software and data bases, the automatic acquisition of the data will reduce input preparation time by orders of magnitude and virtually eliminate input data error. This will substantially reduce the cost of use of visualization tools and make the daily application of such tools in building design and construction attainable.

Task Description

The user loads the space or building description (in form of 3-D building geometry), selects the materials for each surface that affects the rendering, defines the source(s) of light and the associated attributes, selects a view-point, defines the parameters of rendering and executes the simulation.

Example Usage Scenario

The architect has redesigned the space to serve as a computer classroom and has, together with the interior designer, planned the layout of computers and monitors in the space. Since three of the four walls that define this space have large windows, reflection and glare from monitor screens may render this layout unusable. To find the extent of possible reflection and glare, the architect uses a high-end visualization tool to generate a photo-accurate image of the space and all furnishings. The resulting image clearly conveys that reflection from computer monitors will be unacceptable in the current layout. The architect and the interior designer will have to change the orientation (position) of monitors, introduce effective blinds or drapes, or change the layout of the space.

4.12. [XM-2] Project Document Management

Domain process list:

- Project Document Management

4.12.1. Project Document Management

4.12.1.1. Introduction

Overview:

Project Document Management refers to all information pertaining to the documents used to estimate, bid, purchase, and manage the building process as well as for use within the Facilities Management domain. This data identifies the document, the author of the document, changes to the document since the last change, and relationships to other documents.

It has been suggested to the group that the first concentration of our should be on the Contract Drawings represented in the model. It is acknowledged that this is only a small subset of the related documents of the model.

Process Scope:

- Create Drawing View:
- Retrieve Drawing View:

Out-of-Scope:

- All NonCAD Document Views (such as Specifications, Change Orders, etc.)

Definitions:

- **Bulletin** - a collection of Drawings, Specifications, Sketches, and instructions transmitted to the Project Team from the Architect in order to convey a clarification or change to the original drawings issued.
- **Addenda -** Similar to the Bulletin but released by the Architect prior to the signing of a contract between the Owner/Architect and the Construction Team.

- **Drawing** - A 2D representation of a collection of objects that are contained within the model. This may be seen as a view of the model in 2D for a select number of objects within a View Type (such as plan or section).
- **Specification -** A written representation of the objects within the model with instructions on how they are to be constructed (such as materials to be used, techniques in construction, show drawings to be submitted, etc.)

References: Any pertinent references or background materials used

- None at this writing.

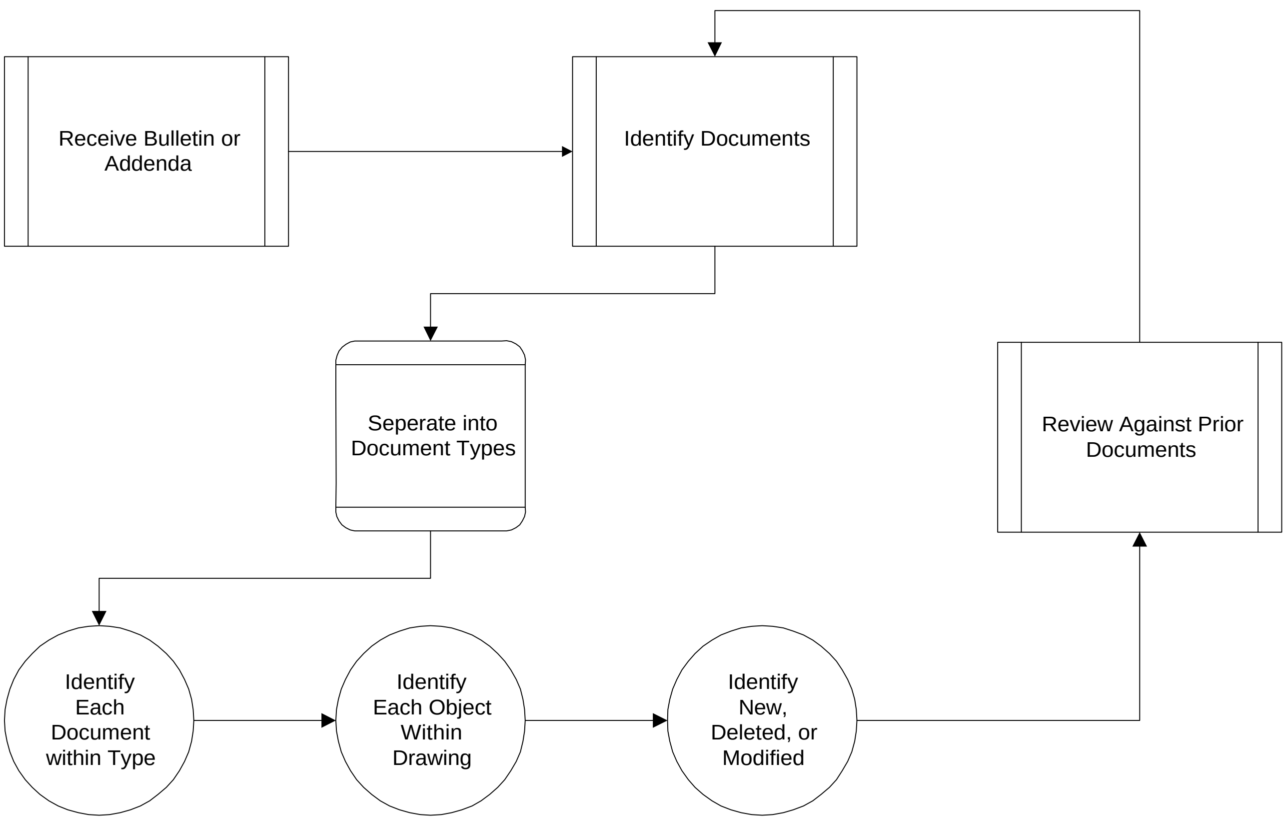
Contributors: The names and chapters of the domain participants

- Raymond H. Brungard NA
- Graham Storer UK
- Arto Kiminieri Nordic
- Richard See NA
- Ken Herold (part time) NA
- Mike Cole (part time) NA

4.12.1.2. Process Diagram – Document Management

The diagram should illustrate how the tasks use the model to get data that was created by previous processes and to store data that may be used by later processes.

See Process Diagrams METAD1.vsd and METAD2.vsd under separate cover.



4.12.1.3. Process Definition – Document Management

4.12.1.3.1. Overview

This section should include overview information about subject process. This overview provides a sentence or two about each bubble in the Process Diagram. The overview should conclude with what bottlenecks or areas of difficulty are frequently encountered in this process, and the benefits of enabling this process through IFC's.

The basic requirement of this process is to be able to create and retrieve views of the model which relate to the objects as 2D drawings used to. This means that a selection of objects may be chosen with a view type (the way in which the objects are to be viewed in 2D, such as plan or section view) to represent a discrete area or areas within the project. These areas can be interpreted as drawings in the sense that they may be printed out or viewed in the same manner as drawings are used today.

Process Task Descriptions

4.12.1.3.2. Create Drawing View

Identify Objects within the model to include in the Drawing View. These objects should be a complete representation of the work for its' view.

Identify the view type used to represent the objects within the drawing. This view type represents the way in which the objects are viewed, usually representing a direction of view, such as plan (viewing from the top).

Provide and apply a reference number, name, revision number, and general information regarding the intended drawing.

Provide for drawing "types", such as plumbing, electrical, concrete, etc.

Provide for additional references for aggregation of information such as Bulletin, Addenda, etc.

4.12.1.3.3. Retrieve Drawing View

Receive the Bulletin, Addenda, or drawing set and their references.

Identify the Drawings within the set.

Identify Drawing type.

Identify the Objects within the drawings

Identify the View of the Objects.

Retrieve additional references.

5. Information Requirements Analysis

5.1. [AR-1] Architectural Model Extensions

5.1.1. Process: Building Shell Design

5.1.1.1. Information Analysis by Task

Please see the process overview description, process diagram and detailed process definition for this process in the "AEC+FM Industry Process Definitions" section of this document.

5.1.1.1.1. Task 1 - Preliminary Building Massing (option 1)

See task description and usage scenario in the Process Definitions section above.

Input Information:

- *A bubble diagram laid out floor by floor (Architecture block & stack)*
- *Structural depths (Structural)*
- *Codes*
- *Core/Circulation*
- *Roof Design*
- *Floor to Ceiling heights*
- *Preliminary BS depths*

Output Information:

- *Refined floor plate shapes (Structural, Architecture)*
- *Refined floor to floor heights (Structural, Architecture)*
- *Volume and massing of the building (Architecture, HVAC, Simulation, Analysis)*
- *Preliminary elevation shape (Architecture)*
- *Exterior Circulation (ramps, balconies, docks, stairs, elevators)*

5.1.1.1.2. Task 2 - Determine Relationship between Shell and Structure

See task description and usage scenario in the Process Definitions section above.

Input Information:

- *Preliminary Massing Studies*
- *Climate*
- *Context*
- *Architectural Styles*
- *Preliminary Design Grid (Architecture)*
- *Preliminary Structural Grid/System (Structural)*

Output Information:

- *Floor plates and design grid (Structural, Architecture)*
- *Refined elevation and model (Architecture)*

5.1.1.1.3. Task 3 - Determine Fenestration (aesthetic criteria)

See task description and usage scenario in the Process Definitions section above.

Input Information:

- *Refined floor plate shapes (Structural, Architecture, Construction)*
- *Refined floor to floor heights (Structural, Architecture)*
- *Preliminary Structural Depths (Structural)*

- *Architectural Styles*
- *Manufactured Systems*
- *Volume and massing of the building (Architecture, HVAC, Simulation, Analysis)*
- *Code requirements (fire access, sill heights, energy)*
- *Preliminary elevation shape (Architecture, Structural)*
- *Building Orientation (Architecture)*

Output Information:

- *Window/Door dimensions (Architecture, HVAC, Simulation, Construction, Analysis)*
- *Window/Door locations (Architecture, HVAC, Simulation, Construction, Analysis)*
- *Glass Area (Architecture, HVAC, Simulation, Construction, Analysis)*
- *Window/Door Type (Architecture (HVAC, Simulation, Construction, Analysis)*
- *Window/Door Framing (Architecture, HVAC, Simulation, Construction, Analysis)*
- *Shading elements (overhang, brise desoleil, landscape elements, Analysis)*

5.1.1.1.4. Task 4 - Define Shell Materials

See task description and usage scenario in the Process Definitions section above.

Input Information:

- *Project Material List (Architecture, Client)*
- *Architectural Context*
- *Lifecycle concerns*
- *Construction Methods (Construction)*
- *Code Considerations*

Output Information:

- *Exterior wall type (HVAC, Simulation, Structural, Construction, Analysis)*
  - *Composition*
  - *Materials*
  - *Connections*
- *Window/Door Type*
  - *Composition*
  - *Materials*
- *Project documents (information to others)*

5.1.1.1.5. Task 5 - Costs

See task description and usage scenario in the Process Definitions section above.

Input Information:

- *Fenestration (Architecture)*
- *Wall type (Architecture)*
- *Window/Door type (Architecture)*
- *Exterior Circulation (ramps, balconies, docks, stairs, elevators)*
- *Preliminary Building Services*
- *Occupancy*
- *Loads (lighting, ventilation)*
- *Waste Stream (greening)*

Output Information:

- *Heat gain numbers*
- *Heat Loss numbers*
- *Preliminary energy analysis*
- *Material*
- *Equipment*
- *Life Cycle Costs/Trade-Offs*
- *Waste Stream/Trade-Offs (greening)*
- *Construction Time*

5.1.1.1.6. Task 6 - Visual Design Refinements

See task description and usage scenario in the Process Definitions section above.

Input Information:

- *Wall Type (Architecture)*
- *Cost*
- *Scale*
- *Building Services*
- *Relationship of Materials*
- *Architectural Style (Architecture)*

Output Information:

- *Details on adornment (Structural, Construction)*

5.1.1.2. IFC Model Impact

This section summarizes the model requirements from all the process tasks analyzed above into two groups Extensions to R1.5.1 model object types and proposed new object types for R2.0.

Extensions to IFC R1.5 object types

- ***IfcWindow***
  Data
  - *{{ Data description type }}*
    - *{{ notes }}*
- ***IfcDoor***
  Data
  - *{{ Data description type }}*
    - *{{ notes }}*

New object types required in IFC R2.0

- ***Parapet IfcWall::Type = parapet***
  Data
  - *{{ Data description type }}*
    - *{{ notes }}*
- ***Snow Build UP***
  Data
  - *{{ Data description type }}*
    - *{{ notes }}*
- ***Wind***
  Data
  - *{{ Data description type }}*
    - *{{ notes }}*
- ***Height***
  Data
  - *{{ Data description type }}*
    - *{{ notes }}*
- ***Function (Handrail/Safety, Screening)***
  Data
  - *{{ Data description type }}*
    - *{{ notes }}*
- ***Louver***
  Data

- *Geometry type*
- *Details type*
- *Material type*
- *Finish type*
- *Free Area (ventilation) type*
- *Screen/mesh size type*
    - *Structural framing for hole (detailed enough???)*

- ***Stair (See Stair Process)***

  Data
  - *{{ Data description type }}*
    - *{{ notes }}*

- ***Ramp***

  Data
  - *Geometry type*
  - *Material type*
  - *Finish type*
  - *Handrail (link) type*
    - *see Handrail*
  - *Guardrail (link) type*
    - *see Guardrail*
  - *Building code (link) type*

- ***Projections (ornamentation) NOTE: better word???***

  Data
  - *Type (Canopy, Flag Pole, gargoyle, prefabricated balcony) type*
  - *Geometry type*
  - *Material type*
  - *Weight type*
  - *Manufacturer type*
  - *Orientation type*
  - *Connections to façade (e.g. bolt, steel clip, etc.) type*

- ***Curtain wall (window wall) (Look at CSI code Uniformat)***

  Data
  - *Assembly type*
  - *Surface (link) type*
  - *Manufacturer type*
  - *Detail type*
  - *Building code (link) type*
  - *Specification (link) type*

- ***Foundation (elements, connections) see foundation design***

***** Note: need to know: floor to floor; floor plate; topography(grade)*****

5.1.1.3. RoadMap Issues

Interoperability Value

Disciplines/Applications from which information is needed:

- *Structural*
- *HVAC*
- *Energy*
- *Codes*

Disciplines/Applications to which information will be supplied:

- *HVAC*
- *Simulation*
- *Construction*
- *Facility Management*
- *Specifications*

Value of software supporting this process

In this section, other domain teams will rank the value of software which supports this process, based on IFC.

- *{{ discipline 1 }} - {{value from 1-10, 1 being the lowest value, 10 being the highest value}}*
- *{{ discipline 2 }} - {{value from 1-10}}*

Sponsor Software Companies

The following organizations that develop software used by AEC end users have shown interest in developing applications which implement the process:

- *{{ company 1 }}*
- *{{ company 2 }}*

5.1.1.4. Issues idendified in reviews

No reviews recorded

Issues

Proposed resolution

- No issues recorded
  - No resolutions recorded

5.1.2. Process: Building Core Design

See Process Definitions section above.

5.1.2.1. Information Analysis by Task

Please see the process overview description, process diagram and detailed process definition for this process in the "AEC+FM Industry Process Definitions" section of this document.

5.1.2.1.1. Task 1 - Determine Core Spaces Needed

See task description and usage scenario in the Process Definitions section above.

Input Information:

- *Space program (owner requirements)*
- *Occupancy (Floor by Floor)*
- *Occupancy Type (Assembly, etc in code)*
- *Codes/Egress (Distances) (Look to AR-2)*
- *Building Services (# and type of service)*
- *Vertical Circulation (#, type)*

Output Information:

- *Spaces (#,type)*

5.1.2.1.2. Task 2 - Determine Core Space Sizes

See task description and usage scenario in the Process Definitions section above.

Input Information:

- *Calculate elevators (Number and sizes length, width, and type (freight vs. passenger)(Freight lobbies)*
- *Calculate Stairs (process #### Length, width)*
- *Floor to Floor Heights*
- *Number of Floors*
- *Calculate Escalator (width, length)*
- *Alarm Stations (width, length)*
- *Restroom Design (process #### length, width, area)*
- *Required spaces (length/width or area) Electrical, Communications, Waste Disposal, Janitorial, Mechanical*

Output Information:

- *Required spaces (length/width or area) (Collection of spaces ie (Core, parking)*

5.1.2.1.3. Task 3 - Layout Core Spaces

See task description and usage scenario in the Process Definitions section above.

Input Information:

- *Structural Grid (grid of object, including shear walls etc.*
- *Max. Distance between exit stairs.*
- *Space efficiency (% usable goal)*
- *Parking Plan*
- *Required spaces (length/width or area) Electrical, Communications, Waste Disposal, Janitorial, Mechanical, Stair, Elevator, Escalator*

Output Information:

- *Core layout (collection of spaces)*

5.1.2.1.4. Task 4 - Detailed Design of Stairs

Covered elsewhere - in Restroom design Process

5.1.2.1.5. Task 5 - Detailed Design of Restrooms

Covered elsewhere - in Restroom design Process

5.1.2.2. IFC Model Impact

This section summarizes the model requirements from all the process tasks analyzed above into two groups Extensions to R1.5.1 model object types and proposed new object types for R2.0.

Extensions to IFC R1.5 object types

- ***None specified***

New object types required in IFC R2.0

- ***Stairs (Actual object)***
  Data
  - *{{ Data description type }}*
    - *{{ notes }}*
- ***Stairs Well***
  Data
  - *{{ Data description type }}*

- *{{ notes }}*

- ***Elevator Shaft***
  Data
  - *{{ Data description type }}*
    - *{{ notes }}*
- ***Elevator***
  Data
  - *{{ Data description type }}*
    - *{{ notes }}*
- ***Emergency services***
  Data
  - *Fire Standpipe type*
  - *Hose type*

5.1.2.3. RoadMap Issues

Interoperability Value

Disciplines/Applications from which information is needed:

- *Structural*
- *HVAC*
- *Telecommunications*
- *Plumbing*
- *Electrical*

Disciplines/Applications to which information will be supplied:

- *Structural*
- *HVAC*
- *Telecommunications*
- *Plumbing*
- *Electrical*
- *Specifications*

Value of software supporting this process

In this section, other domain teams will rank the value of software which supports this process, based on IFC.

- *{{ discipline 1 }} - {{value from 1-10, 1 being the lowest value, 10 being the highest value}}*
- *{{ discipline 2 }} - {{value from 1-10}}*

Sponsor Software Companies

The following organizations that develop software used by AEC end users have shown interest in developing applications which implement the process:

- *{{ company 1 }}*
- *{{ company 2 }}*

5.1.2.4. Issues idendified in reviews

No reviews recorded

Issues

Proposed resolution

- No issues recorded
  - No resolutions recorded

5.1.3. Process: Stair Design

See Process Definitions section above.

5.1.3.1. Information Analysis by Task

Please see the process overview description, process diagram and detailed process definition for this process in the "AEC+FM Industry Process Definitions" section of this document.

5.1.3.1.1. Task 1 - Locate Stairs

See task description and usage scenario in the Process Definitions section above.

Input Information:

- *Configuration (type straight, Scissors)*
- *Owner Requirements*
- *Codes*
- *Occupancy*
- *Circulation*
- *Core Inputs (location exit, etc.)*

Output Information:

- *Location and Type*

5.1.3.1.2. Task 2 - Determine Width

See task description and usage scenario in the Process Definitions section above.

Input Information:

- *Configuration (type straight, Scissors, spiral)*
- *Handrail projection (Depth)*
- *Clear Area (distance) between handrail*
- *Stair use (Fire stair, Ornamental)*
- *Codes (Tread Width(distance))*
- *Egress (# of occupants by building type)*
- *Owner Requirements (Grander defined width)*

Output Information:

- *Width of treads*

5.1.3.1.3. Task 3 - Determine Tread depth and Risers height

See task description and usage scenario in the Process Definitions section above.

Input Information:

- *Floor to Floor Heights*
- *Acoustic rating (stc, impact rating)*
- *Codes (Max and min, ratio, nosing depth)*
- *Owner requirements (Depth, Rise) consistent fall within the ratio*

Output Information:

- *Tread depth*
- *Riser height*
- *Nosing Depth*
- *Landing Locations*

- *Material Type*
- *Finish*

5.1.3.1.4. Task 4 - Determine Landing

See task description and usage scenario in the Process Definitions section above.

Input Information:

- *Stair Width*
- *Acoustic rating (stc, impact rating)*
- *Door, standpipe, handrail, clearance*
- *Special Criteria (depth, width)*

Output Information:

- *geometry of Landings*
- *Material Type*
- *Finish*

5.1.3.1.5. Task 5 - Guardrail Design

See task description and usage scenario in the Process Definitions section above.

Input Information:

- *Code (min. /max height, balustrade spacing, Minimum penetration size)*
- *Special Criteria (min. /max height, balustrade spacing, Minimum penetration size)*

Output Information:

- *Guardrail geometry*
- *Material Type*
- *Finish*
- *Guardrail specifications (min. /max height, balustrade spacing, Minimum penetration size))*

5.1.3.1.6. Task 6 - Handrail Design

See task description and usage scenario in the Process Definitions section above.

Input Information:

- *Code (Minimum projections, min. /max height, diameter, extension from base, extension from top, continuation)*
- *Stair configuration (egress, ornamental)*
- *Special Criteria (Minimum projections, min. /max height, diameter, extension from base, extension from top, continuation)*

Output Information:

- *Handrail geometry*
- *Material Type*
- *Finish*
- *Handrail specifications (Minimum projections, min. /max height, diameter, extension from base, extension from top, continuation)*

5.1.3.1.7. Task 7 - Construction and Materials

See task description and usage scenario in the Process Definitions section above.

Input Information:

- *none identified*

Output Information:

- *none identified*

5.1.3.1.8. Task 8 - Finalize Design

See task description and usage scenario in the Process Definitions section above.

Input Information:

- *Life Safety Requirements*
- *Exits*

Output Information:

- *Lighting needs*
- *Ventilation needs*
- *Pressurization*
- *Signage*
- *Stair Design*

5.1.3.2. IFC Model Impact

This section summarizes the model requirements from all the process tasks analyzed above into two groups Extensions to R1.5.1 model object types and proposed new object types for R2.0.

Extensions to IFC R1.5 object types

- ***None specified***

New object types required in IFC R2.0

- ***Treads***
  Data
  - *RiserHeight type*
  - *TreadDepth type*
  - *TreadMaterial*
  - *NosingMaterial*
  - *TreadType*
- ***Handrails***
  Data
  - *HandrailType type*
  - *Material type*
  - *DepthFromWall type*
- ***Guardrails***
  Data
  - *GuardrailType type*
  - *Material type*
  - *DepthFromWall*
- ***Landings***
  Data
  - *Depth type*
  - *Width type*
  - *Material type*
- ***Stringer***
  Data
  - *Depth type*
  - *Width type*
  - *Material type*
  - *Shape (surfaces) type*

5.1.3.3. RoadMap Issues

Interoperability Value

Disciplines/Applications from which information is needed:

- *Structural*
- *Codes*

Disciplines/Applications to which information will be supplied:

- *Plumbing*
- *Electrical*
- *Codes*
- *Construction*
- *Facility Management*
- *Structural*
- *Specifications*

Value of software supporting this process

In this section, other domain teams will rank the value of software which supports this process, based on IFC.

- *{{ discipline 1 }} - {{value from 1-10, 1 being the lowest value, 10 being the highest value}}*
- *{{ discipline 2 }} - {{value from 1-10}}*

Sponsor Software Companies

The following organizations that develop software used by AEC end users have shown interest in developing applications which implement the process:

- *{{ company 1 }}*
- *{{ company 2 }}*

5.1.3.4. Issues idendified in reviews

No reviews recorded

Issues

Proposed resolution

- No issues recorded
  - No resolutions recorded

5.1.4. Process: Public Restroom Design

See Process Definitions section above.

5.1.4.1. Information Analysis by Task

Please see the process overview description, process diagram and detailed process definition for this process in the "AEC+FM Industry Process Definitions" section of this document.

5.1.4.1.1. Task 1 - Determine Requirements

See task description and usage scenario in the Process Definitions section above.

Input Information:

- *Occupancy type*
- *Program Occupancy (number)*
- *Floor area*
- *Municipal fixtures requirements*
- *ADA (clearances)*
- *Special Criteria (list of fixtures)*

Output Information:

- *Fixtures number and types and spacing(clearance) (urinal, WC wall, WC floor etc.)*

5.1.4.1.2. Task 2 - Layout

See task description and usage scenario in the Process Definitions section above.

Input Information:

- *Fixtures (mounting height, clearances)*
- *Plumbing considerations (Renovation)*
- *Codes (Entry, turnaround space)*
- *accessories (grab bars, mirrors, paper towel, trash, partition etc.) mounting, clearances, width, length, height, depth.*
- *Core constraints (width, length, area, polygonal area)*
- *drainage*
- *Structural Grid*

Output Information:

- *Location, height of fixtures and accessories*
- *Location of walls, doors*
- *Space geometry*
- *FloorDrain*
- *Millwork (cabinets and counter tops)*

5.1.4.1.3. Task 3 - Construction Detailing, Finishes and Lighting

See task description and usage scenario in the Process Definitions section above.

Input Information:

- *Space geometry*
- *Fixtures types, locations*
- *accessories (grab bars, mirrors, paper towel, trash, partition etc.) mounting, clearances, width, length, height, depth.*
- *Client requirements (ie. stone*

Output Information:

- *Partition types*
- *Fixture and accessories manufactures model etc.*
- *** Note: finish decisions ***

5.1.4.2. IFC Model Impact

This section summarizes the model requirements from all the process tasks analyzed above into two groups Extensions to R1.5.1 model object types and proposed new object types for R2.0.

Extensions to IFC R1.5 object types

- ***None specified***

New object types required in IFC R2.0

- ***Fixture (Anything coordinated with another discipline)***
  Data

- *MoutingHeight type*
- *DrainConnectionPoint type*
- *HwConnectionPoint type*
- *CwConnectionPoint type*
- *ElectricalConnectionPoint type*
- *RoughOpening type*
- *Detail (link) type*
- *BuildingCode (link) type*
- *OperatingControlLocation (dispenser conforms to range) type*
- *Material type*
- *Finish type*
- *AssociatedFitting type*
- *GraphicSymbol type*
- *Color type*
- *MountingType type*
- *Manufacturer type*

- ***Accessories (Everything else)***

  Data
  - *MountinHeight type*
  - *BoundingBox type*
  - *RoughOpening type*
  - *Detail (link) type*
  - *BuildingCode (link) type*
  - *OperatingControlLocation (dispenser conforms to range) type*
  - *Material type*
  - *Finish type*
  - *Graphic Symbol type*
  - *Color type*
  - *MountingType type*
  - *Manufacturer type*

- ***Manufactured Partitions***

  Data
  - *Height type*
  - *Width type*
  - *Thickness type*
  - *Door (link) type*
  - *Hardware (link) type*
  - *Material type*
  - *Detail (link) type*
  - *Specification (link) type*
  - *Finish type*
  - *Mounting type*
  - *Manufacturer type*

- ***Millwork (Casework)***

  Data
  - *MountingHeight type*
  - *Geometry type*
  - *MountingHardware type*
  - *Detail (link type)*
  - *Manufacturer type*

5.1.4.3. RoadMap Issues

Interoperability Value

Disciplines/Applications from which information is needed:

- *Structural*
- *Plumbing*
- *HVAC*
- *Electrical*

Disciplines/Applications to which information will be supplied:

- *HVAC*
- *Plumbing*
- *Structural*
- *Electrical*
- *Construction*
- *Facility Management*
- *Specifications*

Value of software supporting this process

In this section, other domain teams will rank the value of software which supports this process, based on IFC.

- *{{ discipline 1 }} - {{value from 1-10, 1 being the lowest value, 10 being the highest value}}*
- *{{ discipline 2 }} - {{value from 1-10}}*

Sponsor Software Companies

The following organizations that develop software used by AEC end users have shown interest in developing applications which implement the process:

- *{{ company 1 }}*
- *{{ company 2 }}*

5.1.4.4. Issues idendified in reviews

No reviews recorded

Issues

Proposed resolution

- No issues recorded
  - No resolutions recorded

5.1.5. Process: Roof Design

See Process Definitions section above.

5.1.5.1. Information Analysis by Task

Please see the process overview description, process diagram and detailed process definition for this process in the "AEC+FM Industry Process Definitions" section of this document.

5.1.5.1.1. Task 1 - Design Roof

See task description and usage scenario in the Process Definitions section above.

Input Information:

- *Budget constraints*
- *Community and regional standards*
- *Environment such as snow or tepid regions*
- *Screening building services*
- *Image (height, patterns, fabric)*
- *Client Requirements (material which would effect pitch)*
- *Lifecycle Requirements*
- *Fire Exiting (penthouse)*
- *Alternative Energy (passive design, orientations, equipment)*
- *Live load based on use*
- *Codes (fire, class, slopes)*
- *Functional requirement (structural loading, pool,)*
- *Building massing*
- *Building materials*
- *Structural*
- *Surrounding Building Scapes*

Output Information:

- *Basic form of roof (i.e. Flat, pitched, shed, etc.)*
- *Material requirements (i.e. clay tile roofing, slate)*
- *Slope*
- *Structural depths*
- *Area of roof planes*
- *Vert/horz projections*
- *Lifecycle*

5.1.5.1.2. Task 2 - Skylight/Clear Story

See task description and usage scenario in the Process Definitions section above.

Input Information:

- *Codes*
- *Environmental*
- *Structural*
- *Client requirements*
- *Day lighting*
- *Ventilation*
- *Lifecycle*
- *Design Intent*
- *Energy requirements*
- *Manufacturer input*

Output Information:

- *Geometry*
- *Glazing properties*
- *Materials (performance properties)*
- *Manufacture information*

5.1.5.1.3. Task 3 - Layout of Services

See task description and usage scenario in the Process Definitions section above.

Input Information:

- *HVAC equipment and piping locations*
- *Telecommunications needs in respect to roof dishes etc.*
- *Plumbing venting stacks*
- *Circulation (stairwell)*
- *Roof Circulation*
- *Amenities (pool, heliport)*

· *Fire Protection*
· *Maintenance requirements*

Output Information:

· *Location and geometry of penetration*
· *Location geometry of loading*
· *Location of amenities and equipment*
· *Equipment access*

5.1.5.1.4. Task 4 - Design Rain/Snow Drainage

See task description and usage scenario in the Process Definitions section above.

Input Information:

· *Structure*
· *Roof geometry*
· *Contributing sources (adjacent surfaces walls etc.)*
· *Geographic location and weather information.*
· *Lifecycle (materials copper vs steel)*
· *Code requirements*
· *Site considerations (drainage)*

Output Information:

· *Water/Snow drainage plan*
· *Rough drain/downspouts location and sizes (interior drainage)*
· *Maintenance requirements*

5.1.5.2. IFC Model Impact

This section summarizes the model requirements from all the process tasks analyzed above into two groups Extensions to R1.5.1 model object types and proposed new object types for R2.0.

Extensions to IFC R1.5 object types

- ***None specified***

New object types required in IFC R2.0

- ***Stairs Well***
  Data
  - *{{ Data description type }}*
    - *{{ notes }}*
- ***Roof***
  Data
  - *Style (flat, sloped, etc.) type*
  - *Fire Classification (A,B, …) type*
  - *Space (link) type*
    - *size and volume for ventilation*
- ***Climate (Check spec and BS)***
  Data
  - *{{ Data description type }}*
    - *{{ notes }}*
- ***RoofSurface***
  Data
  - *Geometry type*
  - *Assembly (material - membrane assembly type*
  - *Lifecycle (link)) type*

- *Surface (link)) type*
- *Specification (link)) type*
- *FireClassification) type*

- ***Surface (Apply to all object)***

  Data
  - *Reflectivity type*
  - *RenderingAttribute (link to material) type*
  - *Color type*
  - *Roughness (list) friction coefficient) type*
  - *Transparency) type*

- ***Specification (property set) (Apply to all object)***

  Data

  - Section (pointer to file or contained text block) type

- ***Assembly (property set) (Apply to all object)***

  Data
  - *Factors (list for assembly) type*
  - *Material (link) (factors/attributes) type*

- ***Lifecycle (property set) (Apply to all object)***

  Data
  - *ServiceLife type*
  - *Maintenance interval type*
  - *Warranty type*
  - *Salvage Value type*
  - *Recyclability (property sets?) type*
  - *Disposal (test field) type*
  - *Cost (link) type*

- ***Skylights (could be domed, barrel vault)***

  Data
  - *Geometry type*
  - *Location type*
  - *Manufacture type*
  - *Glazing type type*
    - *(Ufactor, solor head gain coefficient, vis light transmittance, layers, air space, shading coefficient) (Have someone look at galzing type code/BS)*
  - *GlazingArea) type*
  - *FrameType) type*
  - *Operable (same windows?)) type*
  - *VentilationArea) type*
  - *RoughOpening) type*
  - *FinishedOpening) type*
  - *EdgeType (assembly/detail)) type*
  - *LifeCycle) type*
    - *pulled out to Property set applied to all objects*

- ***Joint (Expansion, Edge condition, Control Joint)***

  Data
  - *Assembly (fill the gap) type*
  - *Type (Expansion, Edge, Control, Score, Reveal) type*
  - *Pointer to objects type*
  - *Details type*
  - *FireRating type*
  - *Waterproof type*
  - *Ventilation type*

- *Manufacturer type*
- *RangeOfMovement type*
- *DirectionOfMovement type*
- *Lifecycle type*

- ***Scupper (General opening/edge and object inserted to take)***
  Data
  - *Geometry type*
  - *Material type*
  - *Detail type*
  - *Manufacturer type*

- ***RoofDrain/DownSpout***
  Data
  - *Detail type*
  - *Location type*
  - *Manufacturer type (text string?)*
  - *Specification (link) type*
  - *Material type*

- ***Gutters***
  Data
  - *Geometry type*
  - *Slope type*
  - *Capacity type*
  - *Detail type*
  - *Interface - Drainage*
  - *FlowVolume type*
  - *TributaryArea (Roof planes, Adjacent Surfaces) type*
  - *PrimaryDrainage type*
  - *Secondard type*
  - *Size type*
  - *Interface - Snow*
  - *SnowZone type*
  - *Load type*

- ***Mech screen***
  Data
  - *Length type*
  - *Width type*
  - *Height type*
  - *Type (assembly) type*

- ***Window cleaning***
  (rigging, tracks, rails, carriage, apparatus, maybe this should be pulled out as a process) (Separate object type?)
  Data
  - *Location type*
  - *Type type*
  - *Connection type*

- ***Projections (mechanical screens)***
  Data
  - *ProjectionType type*
  - *Length type*
  - *Material type*
  - *Weight type*
  - *Orientation type*

- *vertical, horizontal, etc*
- *Connection type*
  - *connection to facade ie. bolt, steel*

- ***Stairs (See Stair Process)***
- ***Access (walkways, etc)***

Data

- *Path type*
- *Composition (assembly) type*
- *Width type*

5.1.5.3. RoadMap Issues

Interoperability Value

Disciplines/Applications from which information is needed:

- *Structural*
- *HVAC*
- *Plumbing*
- *Telecommunications*
- *Electrical*
- *Municipal codes*

Disciplines/Applications to which information will be supplied:

- *Structural*
- *Plumbing*
- *Telecommunications*
- *HVAC (heat gain/heat loss analysis)*
- *Electrical*
- *Municipal Codes*
- *Specifications*

Value of software supporting this process

In this section, other domain teams will rank the value of software which supports this process, based on IFC.

- *{{ discipline 1 }} - {{value from 1-10, 1 being the lowest value, 10 being the highest value}}*
- *{{ discipline 2 }} - {{value from 1-10}}*

Sponsor Software Companies

The following organizations that develop software used by AEC end users have shown interest in developing applications which implement the process:

- *{{ company 1 }}*
- *{{ company 2 }}*

5.1.5.4. Issues idendified in reviews

No reviews recorded

Issues

Proposed resolution

- No issues recorded

- No resolutions recorded

5.2. [AR-2] Compartmentation of Buildings

5.2.1. Process: Compartmentation of buildings

5.2.1.1. Information Analysis by Task

Please see the process overview description, process diagram and detailed process definition for this process in the "AEC+FM Industry Process Definitions" section of this document.

5.2.1.1.1. Task A - Identify Main/Ancillary Use Spaces

See description in the Process definition section above.

Input Information:

- *Project Information*
- *Project Geometry*
- *Building Use Type*
- *Building Geometry*
- *Use Classification*
- *Occupancy*

Output Information:

- *Main Use Spaces*
- *Ancillary Use Spaces*

5.2.1.1.2. Task B - Adjust Main/Ancillary according to Code

See description in the Process definition section above.

Input Information:

- *Project Information*
- *Project Geometry*
- *Building Use Type*
- *Building Geometry*

Output Information:

- *Additional Main Use Spaces if any.*

5.2.1.1.3. Task C - Identify Single Occupancy Spaces

See description in the Process definition section above.

Input Information:

- *Project Information*
- *Project Geometry*
- *Building Use Type*
- *Building Geometry*
- *Use Classification*
- *Occupancy*

Output Information:

- *Single Occupancy Spaces*

5.2.1.1.4. Task D - Check Areas/Volumes to Design Fire Compartments

See description in the Process definition section above.

Input Information:

- *Main Use Spaces*
- *Single Occupancy Spaces.*

Output Information:

- *Fire Compartments.*

5.2.1.2. IFC Model Impact

This section summarizes the model requirements from all the process tasks analyzed above into two groups Extensions to R1.5.1 model object types and proposed new object types for R2.0.

Extensions to existing R1.5.1 Object Types

- ***IfcProject***
- ***IfcSite***
- ***IfcBuilding***
- ***IfcSpace***

New object types required in IFC R2.0

- ***IfcMainUseSpace***
  Data
  - *FireUseClassification IfcClassification*
    - *(A MainUseSpace will have one Fire Use which is assigned from the FireUseClassification)*
- ***IfcAncillaryUseSpace***
  Data
  - *FireUseClassification IfcClassification*
    - *(An AncillaryUseSpace will have one Fire Use which is assigned from the FireUseClassification. An AncillaryUseSpace is contained by one MainUseSpace)*
- ***IfcSingleOccupancySpace***
  Data
  - *SingleOccupancyPossessor STRING*
    - *(Defines who possesses and uses a space for fire compartmentation purposes)*
- ***IfcFireCompartment***
  Data
  - *UseType STRING*
    - *(no description)*

5.2.1.3. RoadMap Issues

Interoperability Value

Disciplines/Applications from which information is needed:

- *Client Brief*
- *Architecture Building Model)*
- *Services Engineers)*

Disciplines/Applications to which information will be supplied:

- *Architecture*

Target Software Companies/Application Type

- *Architects and Fire Officers*
- *CAD systems providers (Autodesk)/Autocad*
- *CAD-support FM applications /space planning, occupancy planning, and asset management databases*

Value of software supporting this process

In this section, other domain teams will rank the value of software which supports this process, based on IFC.

- *Architecture: High (in the top 5)*
- *FM: Very High (in the top 3)*
- *CM/Cost: Very High (in the top 3)*
- *Building Service:*
- *HVAC:*

Sponsor Software Companies

The following organizations that develop software used by AEC end users have shown interest in developing applications which implement the process:

- *Autodesk - UK*
- *SSi*

5.2.1.4. Issues idendified in reviews

No reviews recorded

Issues

Proposed resolution

- No issues recorded
    - No resolutions recorded

5.3. [BS-1] HVAC System Design

5.3.1. Process: HVAC Duct System Design

HVAC Duct System Design supports the design and representation of air distribution ductwork systems. Engineers typically perform these processes during the design phase of a building or project, prior to construction. The process culminates with a set of drawings, schedules, and specifications (construction documents) that can be bid upon and constructed.

5.3.1.1.1. Select and Locate System Components

This step involves selecting and locating the air terminals, boxes (if included in the design), and fans that compose the HVAC duct system.

Input Information:

- *Floor plans*
- *Ceiling grid plans*
- *Reflected ceiling plans*
- *Lighting plans*
- *Structural plans*

- *Sprinkler plans*
- *Piping plans*
- *Smoke detector plans*
- *Speaker plans*

Output Information:

- *Pset_AirTerminal*
- *Pset_CoordinationRequirement*
- *Pset_TerminalBox*
- *IfcPathwayElement*
- *IfcEquipment*

Project Model Usage Requirements:

Object types existing in R1.5.1:

- ***IfcWall***
  Data
  - *All available data*
    - *Location and type information (i.e., fire ratings, special construction types, etc.). This allows the designer to identify the potential locations for air terminals, boxes and fans*
- ***IfcSpace***
  Data
  - *Programme information*
    - *Information specific to the intended function of the space, which is used to determine the number and type of air terminals to be installed.*
- ***IfcCeiling***
  Data
  - *Ceiling Type information*
    - *Information needed to determine the type and location of air terminals to be installed.*
    - *Information needed to determine clearances in interstitial spaces.*
- ***IfcBeam***
  Data
  - *Type, size and location of beams*
    - *This information is needed to prevent conflicts with air terminals and terminal boxes.*
- ***IfcFluidMover***
  Data
  - *All available data.*
    - *This information coupled with the Pset_Fan or Pset_AirHandlingUnit property sets are used to locate and initially specify the fan.*

Object types to add in R2.0:

See the IFC Model Impact section for class details.

5.3.1.1.2. Connect Components with Ducts and Fittings

This step involves using engineering judgment to connect the air terminals, boxes, and fans with ducts and fittings. This information is then used for preparing drawings or specifications which will schematically represent the system under design. These schematics are then used to begin coordination with other disciplines which are impacted by the system.

Input Information:

- *Floor plans*
- *Structural plans*
- *Pset_CoordinationRequirement*

Output Information:

- *Pset_DuctFitting*
- *Pset_DuctSegment*
- *IfcPathwayElement*

Project Model Usage Requirements:

Object types existing in R1.5.1:

- ***IfcWall***
  Data
  - *All available data*
    - *Location and type information (i.e., fire ratings, special construction types, etc.). This allows the designer to locate ducts and fittings appropriately.*
- ***IfcCeiling***
  Data
  - *Ceiling Type information*
    - *Information needed to determine where duct and fittings can be located.*
- ***IfcBeam***
  Data
  - *Type, size and location of beams*
    - *This information is needed to prevent conflicts between beams and ducts and fittings.*

Object types to add in R2.0:

See the IFC Model Impact section for class details.

5.3.1.1.3. Sizing the Duct and Fittings

The sizes of the duct and fittings are calculated.

Input Information:

- *Pset_HVACSpaceElementInformation*
- *Pset_DuctSystemDesignCriteria*
- *Pset_DuctDesignCriteria*

Output Information:

- *Pset_CoordinationRequirement*
- *Pset_RectangularDuctConnection*
- *Pset_RoundDuctConnection*
- *Pset_OvalDuctConnection*

Project Model Usage Requirements:

Object types existing in R1.5.1:

- ***Pset_HVACSpaceElementInformation***
  Data
  - *MaximumAirflow and MinimumAirflow values calculated from the room or space load calculations.*

Object types to add in R2.0:

See the IFC Model Impact section for class details.

5.3.1.1.4. Locate Other System Components

Identify and locate other system components required for the duct system.

Input Information:

- *Pset_DuctSystemDesignCriteria*
- *Pset_DuctDesignCriteria*
- *IfcWall*

Output Information:

- *Pset_CoordinationRequirement*
- *IfcDamper*
- *IfcPathwayElement*
- *IfcControlElement*
- *IfcActuator*

Project Model Usage Requirements:

Object types existing in R1.5.1:

- ***IfcWall***

  Data

  - *All available data*
    - *Location and type information (i.e., fire ratings, special construction types, etc.). This allows the designer to identify locations for system components such as fire dampers.*

Object types to add in R2.0:

See the IFC Model Impact section for class details.

5.3.1.1.5. Interference Check

Identify any interferences with other trades.

Input Information:

- *Plumbing/Sprinkler plans*
- *Piping plans*
- *Floor plans*
- *Ceiling grid plans*
- *Reflected ceiling plans*
- *Lighting plans*
- *Power plans*
- *Structural plans*
- *Pset_CoordinationRequirement*

Output Information:

- *Pset_CoordinationRequirement*

Project Model Usage Requirements:

Object types existing in R1.5.1:

- ***IfcWall***

  Data

  - *All available data*
    - *Location and type information (i.e., fire ratings, special construction types, etc.). This allows the designer to identify conflicts between walls and the duct system.*

- ***IfcBeam***

  Data

  - *Type, size and location of beams*
    - *This information is needed to prevent conflicts with air terminals, terminal boxes, duct, fittings, and equipment.*

Object types to add in R2.0:

See the IFC Model Impact section for class details.

5.3.1.1.6. Identify alternatives to design problems

This step requires the designer to go back and redesign certain portions of the system. This may involve regenerating the schematic design documents and recalculating system component sizes. Note that this step may occur at any point in the process.

Input Information:

- *Pset_CoordinationRequirement*

Output Information:

- *Pset_CoordinationRequirement*

Project Model Usage Requirements:

Object types existing in R1.5.1:

- *None.*

Object types to add in R2.0:

See the IFC Model Impact section for class details.

5.3.1.1.7. Pressure Loss Calculations

Determine the system pressure losses based on the duct system that has been designed.

Input Information:

- *Pset_DuctSystemDesignCriteria*
- *Pset_DuctDesignCriteria*

Output Information:

- *Pset_Fan, Pset_PackagedACUnit*

Project Model Usage Requirements:

Object types existing in R1.5.1:

- ***Pset_Fan, Pset_PackagedACUnit***
  Data
  - *All available data*
    - *Pressure loss performance requirements for the fan or packaged AC unit*

Object types to add in R2.0:

See the IFC Model Impact section for class details.

5.3.1.1.8. Fan Selection

Identify a fan that will appropriately meet the requirements of the duct system.

Input Information:

- *Pset_DuctSystemDesignCriteria*
- *Pset_HVACAirSideSystemDesignCriteria*

Output Information:

- *Pset_CoordinationRequirement*
- *IfcEquipment*
- *Pset_Fan, Pset_PackagedACUnit*
- *Pset_ElectricalCharacteristics*

Project Model Usage Requirements:

Object types existing in R1.5.1:

- ***Pset_HVACAirSideSystemDesignCriteria***
  Data
  - *All available data.*
    - *This information is updated appropriately as the fan system is sized.*
- ***IfcEquipment***
  Data
  - *All available data.*
    - *The information related to the weight and maintenance requirements is updated based on the fan selection.*
- ***Pset_Fan, Pset_PackagedACUnit***
  Data
  - *All available data.*
    - *The information related to the fan or packaged unit is updated with the new performance data from the fan selection.*
- ***Pset_ElectricalCharacteristics***
  Data
  - *All available data.*
    - *The electrical requirements for the selected fan or packaged AC unit.*

Object types to add in R2.0:

See the IFC Model Impact section for class details.

5.3.1.1.9. Generate Final System Representation

This step involves preparing drawings and specifications which will be used as contract documents for bid and construction. These documents complete the design phase of the system.

Input Information:

- *Pset_CoordinationRequirement*

Output Information:

- *Pset_AirTerminal*
- *Pset_TerminalBox*
- *Pset_DuctFitting*
- *Pset_DuctSegment*
- *IfcDamper*
- *IfcPathwayElement*
- *IfcControlElement*
- *IfcActuator*

Project Model Usage Requirements:

Object types existing in R1.5.1:

- *None.*

Object types to add in R2.0:

See the IFC Model Impact section for class details.

5.3.1.2. IFC Model Impact

Usage/Extensions to R1.5.1 object types and property sets

- ***Pset_AirHandlingUnit***
  Data
  - *To be determined*
- ***Pset_HVACAirSideSystemInformation***
  Data
  - *Need to coordinate with duct system design*
- ***Pset_Fan***
  Data
  - *To be determined*
- ***Pset_Insulation***
  Data
  - *Need to coordinate with duct and pipe insulation*
- ***IfcEquipment***
  Data
  - *Need to reconcile relationship with IfcPathwayElement to allow for connectivity*

New object types and property sets required in R2.0

- ***IfcPathwayElement***

  IfcPathwayElement generically connects together parts of a networked system. A networked system can be used to represent a system used to transport fluids, such as a duct or piping system. It can also be used for many other system representations, such as electrical distribution systems, computer networks, etc. Note that some types of pathway elements are part of more than one networked system. For example, a fan powered terminal box participates as part of a duct system as well as an electrical system.

  IfcPathwayElement is a subtype of IfcBuildingElement. This class provides a reference to a PathwayElementType type definition which contains the attributes required for the system being designed. In this manner, a pathway element can have properties of a channel (one input and one output), a junction (many inputs and one output) or a splitter (one input and many outputs).

  NOTE to Modeling Team: The following paragraph is reflective of IFC 1.0 constructs and does not incorporate the IfcNetwork constructs planned for the IFC Release 2.0 Core Model.

  The I_PhysicalConnections interface on IfcElement (from which IfcBuildingElement derives) contains the ConnectionPoints and PointConnections attributes which can be used for collecting physical or logical connections for both nodes and edges. ConnectionPoints can be used to collect pure logical connection points. PointConnections, combined with the information in a connection type property set (i.e., Pset_RectangularDuctConnection, Pset_RoundDuctConnection, Pset_OvalDuctConnection) attached to the referenced IfcPointConnector, provide the required information for the type, size and location of physical connections.

  Data
  - *See the Object Type Definition Tables section for details.*
- ***Pset_RectangularDuctConnection***

  This property set provides size information about a rectangular duct connection.

  Data
  - *See the Object Type Definition Tables section for details.*
- ***Pset_RoundDuctConnection***

This property set provides size information about a round duct connection.

Data

- *See Object Type Definition Tables for details*

- ***Pset_OvalDuctConnection***

  This property set provides size information about an oval duct connection.

  Data

  - *See Object Type Definition Tables for details*

- ***Pset_CoordinationRequirement***

  This property set provides a placeholder for interoperable coordination requirements between different disciplines.

  Data

  - *See the Object Type Definition Tables section for details.*

- ***Pset_AirTerminal***

  This property set will be used by an IfcPathwayElement object for defining Air Terminals.

  Data

  - *See the Object Type Definition Tables section for details.*

- ***Pset_TerminalBox***

  This property set will be used by an IfcPathwayElement object to define Terminal Boxes.

  Data

  - *See the Object Type Definition Tables section for details.*

- ***Pset_DuctFitting***

  This property set will be used by an IfcPathwayElement object to define duct fittings.

  Data

  - *See the Object Type Definition Tables section for details.*

- ***Pset_DuctSegment***

  This property set will be used by an IfcPathwayElement object to define duct segments.

  Data

  - *See the Object Type Definition Tables section for details.*

- ***Pset_DuctDesignCriteria:***

  This property set will typically be used in conjunction with Pset_Fluid and Pset_Insulation.

  Data

  - *See the Object Type Definition Tables section for details.*

- ***Pset_DuctSystemDesignCriteria:***

  This property set will typically be used in conjunction with Pset_Fluid and Pset_Insulation.

  Data

  - *See the Object Type Definition Tables section for details.*

- ***IfcDamper:***

  This object class is a subtype of IfcPathwayElement and is used to define dampers.

  Data

  - *See the Object Type Definition Tables section for details.*

- ***Pset_FireDamper:***

  This property set adds information to an IfcDamper object that is specific to fire dampers.

  Data

  - *See the Object Type Definition Tables section for details.*

- ***Pset_SmokeDamper:***

  This property set adds information to an IfcDamper object that is specific to smoke dampers.

Data

- *See the Object Type Definition Tables section for details.*

- ***Pset_FireSmokeDamper:***

  This property set adds information to an IfcDamper object that is specific to combination fire and smoke dampers.

  Data

  - *See the Object Type Definition Tables section for details.*

- ***Pset_BackdraftDamper:***

  This property set adds information to an IfcDamper object that is specific to backdraft dampers.

  Data

  - *See the Object Type Definition Tables section for details.*

- ***Pset_ControlDamper:***

  This property set adds information to an IfcDamper object that is specific to control dampers.

  Data

  - *See the Object Type Definition Tables section for details.*

- ***Pset_Louver:***

  This property set adds information to an IfcDamper object that is specific to louvers.

  Data

  - *See the Object Type Definition Tables section for details.*

- ***IfcControlElement***

  This class is used to identify control components that are typically a part of any HVAC duct or piping system. The information contained within this class and its related property sets attempt to remain consistent with the BACnet Standard. This allows implementation of the IFC control elements to be compatible with the BACnet Standard as desired.

  BACnet is a very extensive, but not exhaustive specification aimed at providing an interoperable method of generalized Building Control Systems from different vendors. It does provide an object specification, some of which has been integrated into IFC.

  To determine the suitability of the BACnet object attributes required for inclusion in IFC, the BACnet object attributes were categorized into three major groups by the IAI Building Systems domain committee:

  · *External -- Provided by the consultant, design engineer or owner. These are the attributes to be included in the IFC specifications.*

  · *Vendor -- Specifics that depend upon the product offering of the control vendor and the vendor's engineering efforts*

  · *Run-Time – The actual values of the building and systems when under control (values altered by operating staff are considered run-time, not externally specified)*

  The reader is reminded that BACnet is a communication protocol. It is not a database for a building control system, but rather formalized method of communication.

  In order to provide IFC interoperability, the externally specified attributes of the BACnet Objects should be standardized so that design engineers can communicate their requirements to control vendors. All other uses and definitions of the BACnet attributes are defined in the BACnet Specification (ANSI/ASHRAE 135-95).

  Data

  - *See the Object Type Definition Tables section for details.*

- ***IfcActuator***

  This object class subtypes from IfcControlElement to define the various types of actuators.

  Data

  - *See the Object Type Definition Tables section for details.*

- ***Pset_LinearActuator***

  This property set adds information to an IfcActuator object that is specific to linear actuators.

  Data

  - *See the Object Type Definition Tables section for details.*

- ***Pset_RotationalActuator***

  This property set adds information to an IfcActuator object that is specific to linear actuators.

  Data

  - *See the Object Type Definition Tables section for details.*

- ***Pset_ElectricActuator***

  This property set adds information to an IfcActuator object that is specific to electric actuators.

  Data

  - *See the Object Type Definition Tables section for details.*

- ***Pset_PneumaticActuator***

  This property set adds information to an IfcActuator object that is specific to pneumatic actuators.

  Data

  - *See the Object Type Definition Tables section for details.*

- ***Pset_HydraulicActuator***

  This property set adds information to an IfcActuator object that is specific to hydraulic actuators.

  Data

  - *See the Object Type Definition Tables section for details.*

- ***Pset_HandOperatedActuator***

  This property set adds information to an IfcActuator object that is specific to hand operated actuators.

  Data

  - *See the Object Type Definition Tables section for details.*

- ***Pset_Sensor***

  This property set adds information to an IfcControlElement object that is specific to sensors.

  Data

  - *See the Object Type Definition Tables section for details.*

- ***Pset_Controller***

  This property set adds information to an IfcControlElement object that is specific to sensors.

  Data

  - *See the Object Type Definition Tables section for details.*

- ***Pset_ControlElementAnalogInput***

  This property set adds information to an IfcControlElement object that has an analog input. This is a BACnet compatible property set.

  Data

  - *See the Object Type Definition Tables section for details.*

- ***Pset_ControlElementAnalogOutput***

  This property set adds information to an IfcControlElement object that has an analog output. This is a BACnet compatible property set.

  Data

  - *See the Object Type Definition Tables section for details.*

- ***Pset_ControlElementBinaryInput***

This property set adds information to an IfcControlElement object that has a binary input. This is a BACnet compatible property set.

Data

- *See the Object Type Definition Tables section for details.*

- ***Pset_ControlElementBinaryOutput***

  This property set adds information to an IfcControlElement object that has a binary output. This is a BACnet compatible property set.

  Data

  - *See the Object Type Definition Tables section for details.*

- ***Pset_ControlElementMultiStateInput***

  This property set adds information to an IfcControlElement object that has a multi-state input. This is a BACnet compatible property set.

  Data

  - *See the Object Type Definition Tables section for details.*

- ***Pset_ControlElementMultiStateOutput***

  This property set adds information to an IfcControlElement object that has a multi-state output. This is a BACnet compatible property set.

  Data

  - *See the Object Type Definition Tables section for details.*

- ***Pset_ControlElementEventEnrollment***

  This property set adds information to an IfcControlElement object regarding the events that the object participates with. This is a BACnet compatible property set.

  Data

  - *See the Object Type Definition Tables section for details.*

- ***Pset_ControlElementLoop***

  This property set adds information to an IfcControlElement object about the control loop that the object participates with. This is a BACnet compatible property set.

  Data

  - *See the Object Type Definition Tables section for details.*

5.3.1.3. RoadMap Issues

Interoperability Value

Disciplines/Applications from which information is needed:

- *Architectural*
- *Structural*
- *HVAC (Piping plans, thermal loads)*
- *Plumbing/Fire Protection*
- *Electrical*
- *Lighting*

Disciplines/Applications to which information will be supplied:

- *Electrical*
- *HVAC*
- *Plumbing/Fire Protection*
- *Structural*

Value of software supporting this process

In this section, other domain teams will rank the value of software which supports this process, based on IFC.

- *Architecture (7)*
- *Building Services (8)*
- *HVAC (9)*
- *FM (6)*
- *CM/Cost (8)*

Sponsor Software Companies

The following organizations that develop software used by AEC end users have shown interest in developing applications which implement the process:

- *APEC*
- *Carrier*
- *Greenheck*
- *Honeywell*
- *Landis-Staefa*

5.3.1.4. Issues idendified in reviews

No reviews recorded

Issues

Proposed resolution

- No issues recorded
    - No resolutions recorded

5.3.2. Process: HVAC Piping System Design

HVAC Piping System Design supports the design and representation of piping systems. These processes are typically performed by engineers and design-build contractors during the design phase of a building or project, prior to construction. The process culminates with a set of drawings which can be bid upon and constructed.

This section defines the specific requirements for HVAC Piping System Design based on the generalized Building Services System Design described above.

5.3.2.1. Information Analysis by Task

Please see the process overview description, process diagram and detailed process definition for this process in the "AEC+FM Industry Process Definitions" section of this document.

5.3.2.1.1. Select and Locate System Components

This step involves selecting and locating the air terminals, boxes (if included in the design), and fans that compose the HVAC duct system.

Input Information:

- *Floor plans*
- *Ceiling grid plans*
- *Reflected ceiling plans*
- *Lighting plans*
- *Structural plans*

- *Sprinkler plans*
- *Piping plans*
- *Smoke detector plans*
- *Speaker plans*

Output Information:

- *Pset_AirTerminal*
- *Pset_CoordinationRequirement*
- *Pset_TerminalBox*
- *IfcPathwayElement*
- *IfcEquipment*

Project Model Usage Requirements:

Object types existing in R1.5.1:

- ***IfcWall***
  Data
  - *All available data*
    - *Location and type information (i.e., fire ratings, special construction types, etc.). This allows the designer to identify the potential locations for air terminals, boxes and fans*
- ***IfcSpace***
  Data
  - *Programme information*
    - *Information specific to the intended function of the space, which is used to determine the number and type of air terminals to be installed.*
- ***IfcCeiling***
  Data
  - *Ceiling Type information*
    - *Information needed to determine the type and location of air terminals to be installed.*
    - *Information needed to determine clearances in interstitial spaces.*
- ***IfcBeam***
  Data
  - *Type, size and location of beams*
    - *This information is needed to prevent conflicts with air terminals and terminal boxes.*
- ***IfcFluidMover***
  Data
  - *All available data.*
    - *This information coupled with the Pset_Fan or Pset_AirHandlingUnit property sets are used to locate and initially specify the fan.*

Object types to add in R2.0:

See the IFC Model Impact section for class details.

5.3.2.1.2. Connect Components with Ducts and Fittings

This step involves using engineering judgment to connect the air terminals, boxes, and fans with ducts and fittings. This information is then used for preparing drawings or specifications which will schematically represent the system under design. These schematics are then used to begin coordination with other disciplines which are impacted by the system.

Input Information:

- *Floor plans*
- *Structural plans*
- *Pset_CoordinationRequirement*

Output Information:

- *Pset_DuctFitting*
- *Pset_DuctSegment*
- *IfcPathwayElement*

Project Model Usage Requirements:

Object types existing in R1.5.1:

- ***IfcWall***

  Data

  - *All available data*
    - *Location and type information (i.e., fire ratings, special construction types, etc.). This allows the designer to locate ducts and fittings appropriately.*

- ***IfcCeiling***

  Data

  - *Ceiling Type information*
    - *Information needed to determine where duct and fittings can be located.*

- ***IfcBeam***

  Data

  - *Type, size and location of beams*
    - *This information is needed to prevent conflicts between beams and ducts and fittings.*

Object types to add in R2.0:

See the IFC Model Impact section for class details.

5.3.2.1.3. Sizing the Duct and Fittings

The sizes of the duct and fittings are calculated.

Input Information:

- *Pset_HVACSpaceElementInformation*
- *Pset_DuctSystemDesignCriteria*
- *Pset_DuctDesignCriteria*

Output Information:

- *Pset_CoordinationRequirement*
- *Pset_RectangularDuctConnection*
- *Pset_RoundDuctConnection*
- *Pset_OvalDuctConnection*

Project Model Usage Requirements:

Object types existing in R1.5.1:

- ***Pset_HVACSpaceElementInformation***

  Data

  - *MaximumAirflow and MinimumAirflow values calculated from the room or space load calculations.*

Object types to add in R2.0:

See the IFC Model Impact section for class details.

5.3.2.1.4. Locate Other System Components

Identify and locate other system components required for the duct system.

Input Information:

· *Pset_DuctSystemDesignCriteria*
· *Pset_DuctDesignCriteria*
· *IfcWall*

Output Information:

· *Pset_CoordinationRequirement*
· *IfcDamper*
· *IfcPathwayElement*
· *IfcControlElement*
· *IfcActuator*

Project Model Usage Requirements:

Object types existing in R1.5.1:

- ***IfcWall***
  Data
  - *All available data*
    - *Location and type information (i.e., fire ratings, special construction types, etc.). This allows the designer to identify locations for system components such as fire dampers.*

Object types to add in R2.0:

See the IFC Model Impact section for class details.

5.3.2.1.5. Interference Check

Identify any interferences with other trades.

Input Information:

· *Plumbing/Sprinkler plans*
· *Piping plans*
· *Floor plans*
· *Ceiling grid plans*
· *Reflected ceiling plans*
· *Lighting plans*
· *Power plans*
· *Structural plans*
· *Pset_CoordinationRequirement*

Output Information:

· *Pset_CoordinationRequirement*

Project Model Usage Requirements:

Object types existing in R1.5.1:

- ***IfcWall***
  Data
  - *All available data*
    - *Location and type information (i.e., fire ratings, special construction types, etc.). This allows the designer to identify conflicts between walls and the duct system.*
- ***IfcBeam***
  Data
  - *Type, size and location of beams*
    - *This information is needed to prevent conflicts with air terminals, terminal boxes, duct, fittings, and equipment.*

Object types to add in R2.0:

See the IFC Model Impact section for class details.

5.3.2.1.6. Identify alternatives to design problems

This step requires the designer to go back and redesign certain portions of the system. This may involve regenerating the schematic design documents and recalculating system component sizes. Note that this step may occur at any point in the process.

Input Information:

- *Pset_CoordinationRequirement*

Output Information:

- *Pset_CoordinationRequirement*

Project Model Usage Requirements:

Object types existing in R1.5.1:

- *None.*

Object types to add in R2.0:

See the IFC Model Impact section for class details.

5.3.2.1.7. Pressure Loss Calculations

Determine the system pressure losses based on the duct system that has been designed.

Input Information:

- *Pset_DuctSystemDesignCriteria*
- *Pset_DuctDesignCriteria*

Output Information:

- *Pset_Fan, Pset_PackagedACUnit*

Project Model Usage Requirements:

Object types existing in R1.5.1:

- ***Pset_Fan, Pset_PackagedACUnit***

  Data

  - *All available data*
    - *Pressure loss performance requirements for the fan or packaged AC unit*

Object types to add in R2.0:

See the IFC Model Impact section for class details.

5.3.2.1.8. Fan Selection

Identify a fan that will appropriately meet the requirements of the duct system.

Input Information:

- *Pset_DuctSystemDesignCriteria*
- *Pset_HVACAirSideSystemDesignCriteria*

Output Information:

- *Pset_CoordinationRequirement*
- *IfcEquipment*
- *Pset_Fan, Pset_PackagedACUnit*
- *Pset_ElectricalCharacteristics*

Project Model Usage Requirements:

Object types existing in R1.5.1:

- ***Pset_HVACAirSideSystemDesignCriteria***
  Data
  - *All available data.*
    - *This information is updated appropriately as the fan system is sized.*
- ***IfcEquipment***
  Data
  - *All available data.*
    - *The information related to the weight and maintenance requirements is updated based on the fan selection.*
- ***Pset_Fan, Pset_PackagedACUnit***
  Data
  - *All available data.*
    - *The information related to the fan or packaged unit is updated with the new performance data from the fan selection.*
- ***Pset_ElectricalCharacteristics***
  Data
  - *All available data.*
    - *The electrical requirements for the selected fan or packaged AC unit.*

Object types to add in R2.0:

See the IFC Model Impact section for class details.

5.3.2.1.9. Generate Final System Representation

This step involves preparing drawings and specifications which will be used as contract documents for bid and construction. These documents complete the design phase of the system.

Input Information:

- *Pset_CoordinationRequirement*

Output Information:

- *Pset_AirTerminal*
- *Pset_TerminalBox*
- *Pset_DuctFitting*
- *Pset_DuctSegment*
- *IfcDamper*
- *IfcPathwayElement*
- *IfcControlElement*
- *IfcActuator*

Project Model Usage Requirements:

Object types existing in R1.5.1:

- *None.*

Object types to add in R2.0:

See the IFC Model Impact section for class details.

5.3.2.2. IFC Model Impact

Usage/Extensions to R1.5.1 object types and property sets

- ***Pset_AirHandlingUnit***
  Data
  - *To be determined*
- ***Pset_HVACAirSideSystemInformation***
  Data
  - *Need to coordinate with duct system design*
- ***Pset_Fan***
  Data
  - *To be determined*
- ***Pset_Insulation***
  Data
  - *Need to coordinate with duct and pipe insulation*
- ***IfcEquipment***
  Data
  - *Need to reconcile relationship with IfcPathwayElement to allow for connectivity*

New object types and property sets required in IFC R2.0

- ***IfcPathwayElement***

  IfcPathwayElement generically connects together parts of a networked system. A networked system can be used to represent a system used to transport fluids, such as a duct or piping system. It can also be used for many other system representations, such as electrical distribution systems, computer networks, etc. Note that some types of pathway elements are part of more than one networked system. For example, a fan powered terminal box participates as part of a duct system as well as an electrical system.

  IfcPathwayElement is a subtype of IfcBuildingElement. This class provides a reference to a PathwayElementType type definition which contains the attributes required for the system being designed. In this manner, a pathway element can have properties of a channel (one input and one output), a junction (many inputs and one output) or a splitter (one input and many outputs).

  NOTE to Modeling Team: The following paragraph is reflective of IFC 1.0 constructs and does not incorporate the IfcNetwork constructs planned for the IFC Release 2.0 Core Model.

  The I_PhysicalConnections interface on IfcElement (from which IfcBuildingElement derives) contains the ConnectionPoints and PointConnections attributes which can be used for collecting physical or logical connections for both nodes and edges. ConnectionPoints can be used to collect pure logical connection points. PointConnections, combined with the information in a connection type property set (i.e., Pset_RectangularDuctConnection, Pset_RoundDuctConnection, Pset_OvalDuctConnection) attached to the referenced IfcPointConnector, provide the required information for the type, size and location of physical connections.

  Data
  - *See the Object Type Definition Tables section for details.*
- ***Pset_RectangularDuctConnection***

  This property set provides size information about a rectangular duct connection.

  Data
  - *See the Object Type Definition Tables section for details.*
- ***Pset_RoundDuctConnection***

This property set provides size information about a round duct connection.

Data

- *See the Object Type Definition Tables section for details.*

- ***Pset_OvalDuctConnection***

  This property set provides size information about an oval duct connection.

  Data

  - *See the Object Type Definition Tables section for details.*

- ***Pset_CoordinationRequirement***

  This property set provides a placeholder for interoperable coordination requirements between different disciplines.

  Data

  - *See the Object Type Definition Tables section for details.*

- ***Pset_AirTerminal***

  This property set will be used by an IfcPathwayElement object for defining Air Terminals.

  Data

  - *See the Object Type Definition Tables section for details.*

- ***Pset_TerminalBox***

  This property set will be used by an IfcPathwayElement object to define Terminal Boxes.

  Data

  - *See the Object Type Definition Tables section for details.*

- ***Pset_DuctFitting***

  This property set will be used by an IfcPathwayElement object to define duct fittings.

  Data

  - *See the Object Type Definition Tables section for details.*

- ***Pset_DuctSegment***

  This property set will be used by an IfcPathwayElement object to define duct segments.

  Data

  - *See the Object Type Definition Tables section for details.*

- ***Pset_DuctDesignCriteria:***

  This property set will typically be used in conjunction with Pset_Fluid and Pset_Insulation.

  Data

  - *See the Object Type Definition Tables section for details.*

- ***Pset_DuctSystemDesignCriteria:***

  This property set will typically be used in conjunction with Pset_Fluid and Pset_Insulation.

  Data

  - *See the Object Type Definition Tables section for details.*

- ***IfcDamper:***

  This object class is a subtype of IfcPathwayElement and is used to define dampers.

  Data

  - *See the Object Type Definition Tables section for details.*

- ***Pset_FireDamper:***

  This property set adds information to an IfcDamper object that is specific to fire dampers.

  Data

  - *See the Object Type Definition Tables section for details.*

- ***Pset_SmokeDamper:***

  This property set adds information to an IfcDamper object that is specific to smoke dampers.

Data

- *See the Object Type Definition Tables section for details.*

- ***Pset_FireSmokeDamper:***

  This property set adds information to an IfcDamper object that is specific to combination fire and smoke dampers.

  Data

  - *See the Object Type Definition Tables section for details.*

- ***Pset_BackdraftDamper:***

  This property set adds information to an IfcDamper object that is specific to backdraft dampers.

  Data

  - *See the Object Type Definition Tables section for details.*

- ***Pset_ControlDamper:***

  This property set adds information to an IfcDamper object that is specific to control dampers.

  Data

  - *See the Object Type Definition Tables section for details.*

- ***Pset_Louver:***

  This property set adds information to an IfcDamper object that is specific to louvers.

  Data

  - *See the Object Type Definition Tables section for details.*

- ***IfcControlElement***

  This class is used to identify control components that are typically a part of any HVAC duct or piping system. The information contained within this class and its related property sets attempt to remain consistent with the BACnet Standard. This allows implementation of the IFC control elements to be compatible with the BACnet Standard as desired.

  BACnet is a very extensive, but not exhaustive specification aimed at providing an interoperable method of generalized Building Control Systems from different vendors. It does provide an object specification, some of which has been integrated into IFC.

  To determine the suitability of the BACnet object attributes required for inclusion in IFC, the BACnet object attributes were categorized into three major groups by the IAI Building Systems domain committee:

  External -- Provided by the consultant, design engineer or owner. These are the attributes to be included in the IFC specifications.

  Vendor -- Specifics that depend upon the product offering of the control vendor and the vendor's engineering efforts

  Run-Time – The actual values of the building and systems when under control (values altered by operating staff are considered run-time, not externally specified)

  The reader is reminded that BACnet is a communication protocol. It is not a database for a building control system, but rather formalized method of communication.

  In order to provide IFC interoperability, the externally specified attributes of the BACnet Objects should be standardized so that design engineers can communicate their requirements to control vendors. All other uses and definitions of the BACnet attributes are defined in the BACnet Specification (ANSI/ASHRAE 135-95).

  Data

  - *See the Object Type Definition Tables section for details.*

- ***IfcActuator***

  This object class subtypes from IfcControlElement to define the various types of actuators.

  Data

- *See the Object Type Definition Tables section for details.*

- ***Pset_LinearActuator***

  This property set adds information to an IfcActuator object that is specific to linear actuators.

  Data
  - *See the Object Type Definition Tables section for details.*

- ***Pset_RotationalActuator***

  This property set adds information to an IfcActuator object that is specific to linear actuators.

  Data
  - *See the Object Type Definition Tables section for details.*

- ***Pset_ElectricActuator***

  This property set adds information to an IfcActuator object that is specific to electric actuators.

  Data
  - *See the Object Type Definition Tables section for details.*

- ***Pset_PneumaticActuator***

  This property set adds information to an IfcActuator object that is specific to pneumatic actuators.

  Data
  - *See the Object Type Definition Tables section for details.*

- ***Pset_HydraulicActuator***

  This property set adds information to an IfcActuator object that is specific to hydraulic actuators.

  Data
  - *See the Object Type Definition Tables section for details.*

- ***Pset_HandOperatedActuator***

  This property set adds information to an IfcActuator object that is specific to hand operated actuators.

  Data
  - *See the Object Type Definition Tables section for details.*

- ***Pset_Sensor***

  This property set adds information to an IfcControlElement object that is specific to sensors.

  Data
  - *See the Object Type Definition Tables section for details.*

- ***Pset_Controller***

  This property set adds information to an IfcControlElement object that is specific to sensors.

  Data
  - *See the Object Type Definition Tables section for details.*

- ***Pset_ControlElementAnalogInput***

  This property set adds information to an IfcControlElement object that has an analog input. This is a BACnet compatible property set.

  Data
  - *See the Object Type Definition Tables section for details.*

- ***Pset_ControlElementAnalogOutput***

  This property set adds information to an IfcControlElement object that has an analog output. This is a BACnet compatible property set.

  Data
  - *See the Object Type Definition Tables section for details.*

- ***Pset_ControlElementBinaryInput***

  This property set adds information to an IfcControlElement object that has a binary input. This is a BACnet compatible property set.

  Data
  - *See the Object Type Definition Tables section for details.*

- ***Pset_ControlElementBinaryOutput***

  This property set adds information to an IfcControlElement object that has a binary output. This is a BACnet compatible property set.

  Data
  - *See the Object Type Definition Tables section for details.*

- ***Pset_ControlElementMultiStateInput***

  This property set adds information to an IfcControlElement object that has a multi-state input. This is a BACnet compatible property set.

  Data
  - *See the Object Type Definition Tables section for details.*

- ***Pset_ControlElementMultiStateOutput***

  This property set adds information to an IfcControlElement object that has a multi-state output. This is a BACnet compatible property set.

  Data
  - *See the Object Type Definition Tables section for details.*

- ***Pset_ControlElementEventEnrollment***

  This property set adds information to an IfcControlElement object regarding the events that the object participates with. This is a BACnet compatible property set.

  Data
  - *See the Object Type Definition Tables section for details.*

- ***Pset_ControlElementLoop***

  This property set adds information to an IfcControlElement object about the control loop that the object participates with. This is a BACnet compatible property set.

  Data
  - *See the Object Type Definition Tables section for details.*

5.3.2.3. RoadMap Issues

Interoperability Value

Disciplines/Applications from which information is needed:

- *Architectural*
- *Structural*
- *HVAC*
- *Plumbing/Fire Protection*
- *Electrical*

Disciplines/Applications to which information will be supplied:

- *Electrical*
- *HVAC*
- *Plumbing/Fire Protection*
- *Structural*

Value of software supporting this process

In this section, other domain teams will rank the value of software which supports this process, based on IFC.

- *Architecture (7)*
- *Building Services (8)*
- *HVAC (9)*
- *FM (6)*
- *CM/Cost (8)*

Sponsor Software Companies

The following organizations that develop software used by AEC end users have shown interest in developing applications which implement the process:

- *APEC*
- *Carrier*
- *Greenheck*
- *Honeywell*
- *Landis-Staefa*

5.3.2.4. Issues idendified in reviews

No reviews recorded

Issues

Proposed resolution

- No issues recorded
    - No resolutions recorded

5.4. [BS-3] Pathway Design and Coordination

5.4.1. Process: Pathway Design and Coordination

The design of pathways contains the draft layout, the coordination and the representation of mechanical and electrical system-pathways to be installed.

This design process is carried out after the first coordination with the architect and structural engineers, and includes load estimates, energy and systems definitions required for a building.

The process ends with drawings containing the coordinated pathways for the mechanical and electrical installations (i.e. heating, cooling, air-conditioning, plumbing, fire-protection and electrical power) within a building.

Based on the building model and the conditions (program) defined by the customer, an initial estimate of required energy, technical equipment and systems is defined. The process of designing the pathway starts by defining the required spatial extents for technical equipment, piping, ducting and electrical routes.

A rough building layout by the architect will frequently be available showing the suggested locations for plant rooms and risers.

Considering these parameters, the engineer defines the necessary locations for plant areas and suggests the routing of the main pathways.

The required plant area and main pathways are represented in the M & E drawings.

This draft is presented to the architect/customer with details on space requirements (sections). Thereafter, a review of the suggested design solution will take place, taking into account the structure, the initial and future investment, user requirements, operating expenses and the flexibility achieved.

Parameters from the building model, the definition of systems and the routes of each media type can be combined to define the pathway. Air ducts, including equipment (fire dampers, VAV-boxes, etc.) are combined to form a ventilation pathway. Pipes for heating, cooling or plumbing are combined to form a media pathway. Electrical trays are combined to form an electrical pathway. Each pathway should allow variables for necessary insulation or fire proofing, as well as variables for necessary access for installation and maintenance. The optimization of the pathway itself can be done by varying the distance and position of ducts, pipes or trays. Every pathway must be coordinated within the architectural and structural restraints, as well as with each other.

A final definition of the spatial requirements for technical equipment and media distribution, defines the location of the pathway. The translation of the pathway into geometrical forms is carried out. These drawings serve as a guideline for the ongoing building services design.

The definitions of the structural systems (flat slab, concrete or steel construction, beams, etc.) reflect the location of the plant areas, risers and pathways. Collision detection with walls, slabs, binding beams etc. should be made and openings have to be defined.

5.4.1.1. Information Analysis by Task

Please see the process overview description, process diagram and detailed process definition for this process in the "AEC+FM Industry Process Definitions" section of this document.

5.4.1.1.1. Task A - Defining required space for stations

This step contains the dimensioning of main components for different systems, inquiry of the approximate space requirement and corresponding placing of the technical areas in the building model. See also - description in the Process definition section above.

Input Information:

- *system design criteria*
- *building model from architect*
- *load calculations and media/system type*
- *preliminary systems definition*
- *other requests like accessibility, flexibility, installation, maintenance*
- *coordination with client*

Output Information:

- *needed plant area and room for the central supply*
- *placing of the central supply in the building*

5.4.1.1.2. Task B - Defining the required space for pathways

This step contains the dimensioning of the energy and media supply as well as the specification of *pathway.* See also - description in the Process definition section above.

Input Information:

- *system design criteria*
- *building model from architect*
- *loads and required medium*
- *appropriate system*
- *other items like accessibility, insulation, fire proofing, installation*

Output Information:

- *required area or volume of pathway*

5.4.1.1.3. Task C - Geometrical representation of stations and pathways

This step contains the geometrical representation of the defined centralized media supply and pathway. See also - description in the Process definition section above.

Input Information:

- *areas for technical plant equipment*
- *placement of plant equipment*
- *space requirement of pathway*

Output Information:

- *building model with geometrical representation of technical plant equipment and pathways.*

5.4.1.1.4. Task D - Interference check

Collision detection with other technical services and the building model . See also - description in the Process definition section above.

Input Information:

- *building model with geometrical representation of technical plant equipment and pathways*
- *coordination requirement*

Output Information:

- *coordination requirement*

5.4.1.1.5. Task E - Identify alternatives to resolve the collisions

This step requires the designer to go back and redesign certain portions of the system. This may involve regenerating the schematic design documents and recalculating system component. Note that this step may occur at any point in the process. See also - description in the Process definition section above.

Input Information:

- *coordination requirement*

Output Information:

- *geometrical representation of pathway*

5.4.1.1.6. Task F - Itemization of Pathway

This step contains the detailed output of a pathway. By consideration of departures and branching as well as the location and distance of each pipe or duct the cross-sectional dimension of the pathway is brought into line with the respective conditions and will be optimized. See also - description in the Process definition section above.

Input Information:

- *coordination requirement*
- *thermal load calculations*
- *building model from architect*

Output Information:

- *geometrical representation of pathway*
- *coordination requirement*

5.4.1.1.7. Task G - Coordination of branches

This step contains the coordination of different trades within the design of pathway at branchings as well as the coordination with structural conditions like binding beams etc. See also - description in the Process definition section above.

Input Information:

- *geometrical representation of pathway*
- *coordination requirement*

- *building model from architect*

Output Information:

- *coordinated allocation scheme of pathway*

5.4.1.1.8. Task H - Interference check

Collision detection with other disciplines and building model . See also - description in the Process definition section above.

Input Information:

- *building model geometrical representation of technical facilities and pathways*
- *cost estimating and operating expanses for HVAC-Systems*
- *building model*
- *coordination requirement*

Output Information:

- *coordination requirement*

5.4.1.1.9. Task I - Determination of openings

This step contains the specification of openings. See also - description in the Process definition section above.

Input Information:

- *coordination requirement*
- *geometrical representation of pathway*

Output Information:

- *coordination requirement*
- *size and placement*

5.4.1.1.10. Task J - Generate final system

This step contains the design of drawings or specifications which are used as a basis for further systems design. See also - description in the Process definition section above.

Input Information:

- *coordination required*

Output Information:

- *see definitions of objects and attributes of*
  - *AR-1 Completion of Architectural Model*
  - *BS-1 HVAC System Design*
  - *BS-2 Power and Lighting Systems Design*
  - *ST-1 Steel Frame Structures*
  - *ST-2 Reinforced Concrete Structures*

5.4.1.2. IFC Model Impact

This section summarizes the model requirements from all the process tasks analyzed above into two groups Extensions to R1.5.1 model object types and proposed new object types for R2.0.

Extensions to existing R1.5.1 Object Types

The coordination of the pathways deals with existing definitions of objects and attributes in the following projects :

- *AR-1 Completion of Architectural Model*
- *BS-1 HVAC System Design*
- *BS-2 Power and Lighting Systems Design*
- *ST-1 Steel Frame Structures*

· *ST-2 Reinforced Concrete Structures*

The basic information as well as the results of the coordination take effect in the processes listed above. For further information please refer to these documents.

New object types required in IFC R2.0

- ***None defined***

5.4.1.3. RoadMap Issues

Interoperability Value

Disciplines/Applications from which information is needed:

- *Architectural*
- *Structural*
- *HVAC*
- *Plumbing / Fire Protection*
- *Electrical*
- *Lighting*

Disciplines/Applications to which information will be supplied:

- *Architectural*
- *Structural*
- *HVAC*
- *Plumbing / Fire Protection*
- *Electrical*
- *Lighting*
- *Cost Estimating*
- *Facility Management*

Value of software supporting this process

In this section, other domain teams will rank the value of software which supports this process, based on IFC.

- *Note assessed*

Sponsor Software Companies

The following organizations that develop software used by AEC end users have shown interest in developing applications which implement the process:

- *RoCAD Informatik*
- *PHi-Tech*
- *GTS*
- *‘ESS*
- *Ziegler Informatics*
- *RoCAD Informatik*
- *Triplan GmbH*
- *Pit-cup GmbH*

5.4.1.4. Issues idendified in reviews

No reviews recorded

Issues

Proposed resolution

- No issues recorded
  - No resolutions recorded

5.5. [BS-4] HVAC Loads Calculation

5.5.1. Process: Building Heating and Cooling Load Calculation

Load calculations serve as the basis for all design stages of the building services design. The results of the load calculations enable the designer to dimension the plant equipment and to determine the required space for plant room.

Load calculations are an official proofing method in Germany for example the proof for heat loss protection must be given in the course of a project), a mode for calculating the heating cooling load or for the yearly dynamic load simulation:

The process terminates in the complete calculations and the data exchange into the IFC model.

The chapter on hand defines the prerequisites for the computer-aided load calculation using of the thermal building model (refer to VDI Guideline 6021, green paper).

After the completion of the building model with its geometric and physical building specifications by the architect, the data is to be extracted using the Aspect Model Load Calculations -- The Thermal Building Model. The thermal building model includes all architectural building components of a defined room, the attributes and the relationships of the components to each other. The thermal building model does not include any the description of the neighboring buildings (e.g. input for external shading).

The parameters like the room temperatures, required air changes, people or machine loads or other necessary data is submitted if known to the design team. If certain data is not know to the design team plausible data is assumed to provide preliminary answers.

The data exchange to the thermal building model does not require any exchange of the graphical data. The thermal building model is independent from the calculation method applied because it describes only the physical data.

After the exchange of data, the engineer checks data transmitted for completeness and possibly amend the data. The engineer has to input the boundary conditions as well as the meteorological data for the load calculation method.

The definition of zones, as a result of the assigned plant equipment, can be carried out by simply numbering them. All rooms of one level having common boundaries can be defined as one zone. Another form of zoning can be made by direct plant assignment. This method ensures, that considerations of energy as well as the simultaneity of use conditions within plants are considered.

As a results of load calculations, the physical qualities of building components may be changed and submitted to an optimization process. This is requested to the IFC-building model. After changing the corresponding data a further exchange of basic data is carried out and the process starts once more.

A revision phase is necessary if there is change to the plant assignment or there are variations to the boundary conditions within the process.

At the end of the process the results of the load calculations are provided for the IFC model for further processing. The definition of technical stations, pathway and their space requirements as well as the dimensioning of system components for building services design are based on these results.

5.5.1.1. Information Analysis by Task

Please see the process overview description, process diagram and detailed process definition for this process in the "AEC+FM Industry Process Definitions" section of this document.

5.5.1.1.1. Task A - IFC-Model take-over

This step contains the import of extracted data from the building model like component geometry and component qualities. The construction of this physical data exchange format corresponds in the construction to a Physical-STEP file. See also - description in the Process definition section above.

Input Information:

- *All structural components of a room, referring to the aspect model of the Thermal Building Model*
- *All component parameters of the structural components (thermal storage)*
- *The relationship of structural components with each other or the outside area.*
- *Alternatively usage conditions*

Output Information:

- *Preparation of data*

5.5.1.1.2. Task B - Specification of zones

See description in the Process definition section above.

Input Information:

- *Building geometry*
- *Use conditions*
- *Plant assignment*

Output Information:

- *Preparation of data*

5.5.1.1.3. Load calculations

This step contains the execution of the load calculations. See also - description in the Process definition section above.

Input Information:

- *Data from the thermal building model*
- *Data preparation and specification of zones*
- *Use conditions*
- *Meteorological and thermal boundary conditions*

Output Information:

- *Room-, zones-(plants-) and building wise load calculations as*
- *Energy consumption proof*
- *Heating load*
- *Cooling load*
- *Annual energy requirement*
- *Building simulation*
- *Requirements on the building*

5.5.1.1.4. Task D - Results into IFC-Model

Exchanging the results of the load calculations to the IFC model. See also - description in the Process definition section above.

Input Information:

- *Detailed load calculations*

Output Information:

- *Results of calculation in abridged version*

5.5.1.1.5. Task E - Design modifications

This step contains the iterative event for the execution of all calculations by variation or change of the zones, usage requirements etc., according to optimization by changing parameters. See also - description in the Process definition section above.

Input Information:

- *Change of zone division (plant assignment)*
- *Change of use conditions*
- *Change of boundary conditions*

Output Information:

- *Calculating variants*
- *New detailed load calculations*
- *New calculation results (requirements on the building model)*

5.5.1.2. IFC Model Impact

This section summarizes the model requirements from all the process tasks analyzed above into two groups Extensions to R1.5.1 model object types and proposed new object types for R2.0.

Extensions to existing R1.5.1 Object Types

- attributes	*comments*	
• *IfcProject*		
- project short name (mark)	*ProjectReference*	*IfcString*
- project name		
- client		
- building service engineer		
- created by		
- revision number		
- comment		
• *IfcBuilding*		
- referenced project	*ref(project)*	
- building short name		
- building name		
- ground touching floor area	*m2*	
- ground-level - NNG	*m NN*	
- building height - h	*m*	
• *IfcBuildingStorey*		
- referenced building	*ref(building)*	
- storey short name		
- storey name		
- storey height (of floor construction)	*m NN*	

New object types required in IFC R2.0

- attributes	*comments*
• *IfcFunctionalUnit (Zone)*	
- referenced building	*ref(building)*
- name of functional unit	
• *IfcRoom*	
- referenced storey	*ref(storey)*
- referenced functional unit	*ref(functional unit)*
- room short name	
- room name	
- room temperature	*oC*

- *not full conditioned*	*Y/N*
- *storey height for room*	*m*
- *room height*	*m*
- *floor level (of floor finish)*	*m*
- *room perimeter*	*m*
- *room ground area*	*m2*
- *room volume*	*m3*

- ***Structural Components (general Type)***

- *structure component type number (index)*	*unique human interpretable number*
- *structure component type name*	
- *infiltration coefficient (Window)*	*m3/(mhPa2/3)*
- *airflow between layers*	*Y/N NT*

- ***Non Heat Storing Structural Components***

- *structural component number (index)*	*unique human interpretable number*
- *structural component type name*	
- *heat transmission coefficient*	*W/m2K*
- *radiation transmission coefficient - glazing*	*b-value*
- *grade of energy flow through the component*	
- *airflow through joints*	*m3/(hPa2/3)*

- ***Heat Storing Structural Components***

- *structural component name*	
- *heat conducting coefficient*	*W/mK*
- *thickness of the layer*	*m*
- *density of the layer*	*kg/m3*
- *lower value of diffusion coefficient*	
- *upper value of diffusion coefficient*	
- *specific heat capacity of the layer*	*kJ/kgK*

- ***Structural Components***

- *referenced type*	*ref(structural component type)*
- *structural component number*	
- *structural component orientation*	
- *from true north*	*in degree*
- *structural component slope*	*in degree*
- *structural component width*	*in m*
- *structural component height*	*in m*
- *structural component area*	*m2*

- ***Specific - for Non Heat Storing Structural Components***

- *referenced type*	*ref(structural component type for non heat storing components)*
- *number of horizontal joints*	
- *number of vertical joints*	
- *length of all joints*	*m*
- *radiation transmission coefficient of the outside sun protection devices*	*b- value*
- *radiation transmission coefficient of the indoor sun protection devices*	*b- value*
- *window projection length b*	*in m*
- *window projection length d*	*in m*
- *window projection length f*	*in m*
- *window projection length c*	*in m*
- *glass area fraction*	

- ***Room Usage Parameters***

- *referenced room*	*ref(room)*
- *usage unit*	*see below*
- *maximum value*	*see below*
- *usage grade*	*see below*

- *constant*	*Y/N*
- *value until 1 o'clock*	*%*
- *and so on*	*%*
- *value until 24 o'clock*	*%*

comment: the usage units are described in the following table:

- *usage unit*	*mark*	*maximum value*	*using grade*
- *persons*	*P*	*number*	*activity (1,2,3)*
- *lighting*	*B*	*W*	*room load factor*
- *machines*	*M*	*W*	*convective component*
- *air supply*	*ZU*	*temperature in oC*	
- *outside air*	*AU*	*mass flow in kg/s*	
- *air extraction*	*AB*	*temperature of the incoming air in oC*	
- *desired room temperature*	*RT*	*temperature in oC*	
- *air change rate*	*LW*	*1/ h*	
- *heat supply or removal*	*S*	*W*	

- ***Room to Structural Component - Relation***

- *referenced structural component*	*ref(structural component)*
- *referenced room*	*ref(room)*
- *structural component index*	*predefined list of indices*
- *orientation*	*front or rear*

- ***Additional information - FOR GERMANY ONLY***
- ***GEB / WSV - building data according to DIN 4701 and WSchV´(heat loss regulation)***

- *referenced building*	*ref(building)*
- *building type*	*E = single house*
	R = multiple house
- *situation*	*N = normal*
- *F = free (windy)*	
- *kind of building*	*N = normal inside temperature*
	G = lower level inside temperature
- *building*	*as defined in WSchV*
- *building between others*	*Y/N*
- *relation l/b from floor slab areas*	*as defined in DIN 4701*

5.5.1.3. RoadMap Issues

Interoperability Value

Disciplines/Applications from which information is needed:

- *Architectural*

Disciplines/Applications to which information will be supplied:

- *HVAC Pathway Design and Coordination*
- *HVAC Duct System Design*
- *HVAC Hydronic System Design*
- *Cost Estimating*
- *Structural*
- *Architectural*

Value of software supporting this process

In this section, other domain teams will rank the value of software which supports this process, based on IFC. Values from 1-10, 1 being the lowest value, 10 being the highest value

- *Not assessed for this process*

Sponsor Software Companies

The following organizations that develop software used by AEC end users have shown interest in developing applications which implement the process:

- *None documented*

5.5.1.4. Issues idendified in reviews

No reviews recorded

Issues

Proposed resolution

- No issues recorded
    - No resolutions recorded

5.6. [CS-1] Code Checking - Energy Codes

5.6.1. Process: Commercial and Residential Energy Code Compliance Checking

5.6.1.1. Information Analysis by Task

Please see the process overview description, process diagram and detailed process definition for this process in the "AEC+FM Industry Process Definitions" section of this document.

The processes illustrated above will be employed in code checking applications that address the following codes:

1. ASHRAE/IESNA Standard 90.1-1989 (Std 90.1)
2. Model Energy Code (MEC - all recent years)

The specific tasks illustrated in the diagrams above are all embedded within existing widely-distributed applications.

5.6.1.1.1. All Tasks

See description in the Process definition section above.

The inputs and outputs of the individual process tasks are not generally shared with other applications and are too numerous to be conveniently listed as separate task inputs and outputs using this format. For now, they have simply be summarized for the entire process below. Use of existing classes has not been noted, except where new attributes are required. Because the product model usage requirements are not broken down by task, they are identical to IFC Model Impact section, and the information is shown once there. [This information is also shown in an accompanying spreadsheet table.]

Input Information:

- *Code Requirements*
- *Building Model*
- *Analysis Rules*

Output Information:

- *Object Constraints*
    - *IfcPropetyConstaints*
    - *IfcIntent*
- *IfcAgregateControl*
- *Code Violation Reports*
- *Compliance Performance Results*
- *Compliance Reports*

5.6.1.2. IFC Model Impact

This section summarizes the model requirements from all the process tasks analyzed above into two groups Extensions to R1.5.1 model object types and proposed new object types for R2.0.

Extensions to existing R1.5.1 Object Types

- ***IfcLayeredElement (Interfaces added to core class)***

  Data
  - *ContinuousRvalue - Ifcreal*
    - *Continuous R-value of assembly including air films, cladding, gypsum board and sheathing layers*
  - *AssemblyUfactor - Ifcreal*
    - *Overall assembly U-factor*

- ***IfcMaterialLayerSet (Interfaces added to resource schema)***

  Data
  - *ContinuousRvalue - Ifcreal*
    - *Continuous R-value of assembly including air films, cladding, gypsum board and sheathing layers*
  - *AssemblyUfactor - Ifcreal*
    - *Overall assembly U-factor*
  - *ParallelLayer1Material - Ref[IfcMaterialLayer]*
    - *Reference to first part of parallel portion of layered assembly*
  - *ParallelLayer2Materiall - Ref[IfcMaterialLayer]*
    - *Reference to second part of parallel portion of layered assembly*
  - *AspectRatioOfLayers - Ifcreal*
    - *Ratio of layer 1 to layer 2*

- ***IfcMaterialLayer (Interfaces added to resource schema)***

  Data
  - *MaterialType - Ref[IfcMaterialTypeLibraryEntry]*
    - *Type reference for homogenous material -- only used if not a material set*
  - *MaterialSet - Ref[IfcMaterialLayerSet]*
    - *Set of materials for material -- only used if not a homogenous material*

- ***IfcWall (Attribute added to core class)***

  Data
  - *AboveGrade - IfcReal*
    - *Ratio of wall area that is above grade to total wall area*

- ***IfcRoof (This class to replace IfcRoofSlab because there are several other roof types)***

  Data
  - *GenericType - IfcRoofTypeEnum*
    - *Predefined generic types are specified in an Enum. A Type definition is available for each generic type (as the required attributes differ). Use TypeDefinition corresponding to this generic type.*
  - *RoofType - Ref[IfcfTypeDefinition]*
    - *Reference to a type definition that links to attributes defining the element (either shared by all instances or added to the ExAttributeSets). Specific TypeDef determined by the Generic Type above.*

- ***IfcFillingElement***
  Data
  - *FillingElementType - [Ref [IfcFillingElementTypeLibraryEntry]*
    - *Predefined generic filling element types specified in a library*
  - *ProjectionFactor - IfcReal*
    - *The ratio of shading projection depth to the height of window*

New object types required in IFC R2.0

- ***IfcPropertyConstraint (To establish a specific limit on an object or attribute of an object)***
  Data
  - *Source - IfcOwnerId*
    - *Code/Standard reference*
  - *ReferenceObject - IfcProjectObject/ IfcAttributeObject*
    - *Object / attribute reference for which the constraint is specified*
  - *Relation - IfcNumericRelation*
  - *ConstraintType - IfcConstraintLevel*
  - *NoticeText - IfcString*
- ***IfcIntent (A collection of attributes representing design intent)***
  Data
  - *Source - IfcOwnerId*
    - *Code/Standard reference ??*
  - *Description - IfcString*
    - *Description of the code requirement*
- ***IfcAggregateControl (A collection of attributes representing the logical relationships between design intent and constraint)***
  Data
  - *Source - IfcOwnerId*
    - *Code/Standard reference*
  - *Operation - IfcLogicalOperation*
    - *Logical relationship between intent and constraint*
- ***IfcBuildingEnvelope***
  Data
  - *AggregateOf - Set[0:N] Ref[IfcLayeredElement]*
    - *Contains references to all instances of layered elements which form the envelope*
  - *OccupancyType - IfcEnvelopeOccupancyTypeEnum*
    - *Envelope occupancy type according to the Standard*
  - *InternalLoadDensity - IfcReal*
    - *Total internal load based on the occupancy*
  - *ThermalLoad - IfcReal*
    - *Envelope load based on the proposed design*
- ***IfcSkylight***
  Data
  - *GenericType - IfcSkylightTypeEnum*
    - *Predefined generic types are specified in an Enum. A Type definition is available for each generic type (as the required attributes differ). Use TypeDefinition corresponding to this generic type.*
  - *SkylightType - Ref[IfcTypeDefinition]*
    - *Reference to a type definition which links to attributes defining the element (either shared by all instances or added to the ExAttributeSets). Specific TypeDef determined by the GenericType above.*
- ***IfcLightingElement (An aggregation class containing all the lighting fixtures)***
  Data
  - *ReferenceObjects - Ref[IfcFixture]*

  - *Contains references to all instances of IfcFixture that are part of the lighting system*
- *OccupancyType - IfcLightingOccupancyType*
  - *Lighting occupancy type according to the Standard*
- *LightingPowerDensity - IfcReal*
  - *Lighting power density specified by the Code (based on Occupancy type)*
- *LighingPower - IfcReal*
  - *Total lighting power for the proposed design*

- ***IfcLightingFixture***

  Data
  - *Category - Ref[IfcLightingFixtureType LibraryEntry]*
    - *The category of lighting fixture*
  - *NumberOfLampsPerFixture - IfcReal*
    - *Number of lamps per fixture*
  - *FixtureIdentification - IfcString*
    - *Fixture identification on plan*
  - *FixtureWattage - IfcInteger*
    - *Total input wattage of the fixture including lamps and ballast*
  - *NumberOfFixtures - IfcInteger*
    - *Total number of this fixture type used in the building*

- ***IfcMaterial Type (Class structure for material properties library--Not addition to Core class)***

  Data
  - *Type - IfcMaterialTypeEnum*
    - *Describes the function of the material layer as an Enum*
  - *ThermalResistance - IfcReal*
    - *Thermal resistance of the material for unit thickness*
  - *HeatCapacity - IfcReal*
    - *Specific heat capacity of the wall material*

- ***IfcFillingElementType (Class structure for filling element library--Not addition to Core class)***

  Data
  - *FramingType - IfcFrameTypeEnum*
    - *Enum representing the frame type*
  - *GlazingType - IfcGlazingTypeEnum*
    - *Enum representing the glazing type*
  - *ThermalResistance - IfcReal*
    - *Thermal resistance of the filling material*
  - *ShadingCoefficient - IfcReal*
    - *Shading coefficient of filling material*

- ***IfcLightingFixtureType (Class structure for lighting fixture library--Not addition to Core class)***

  Data
  - *Description - IfcString*
    - *Description of the lighting Fixture*
  - *LampType - IfcLampTypeEnum*
    - *Lamp type*
  - *LampDescription - IfcString*
    - *Description of the lamp type*
  - *WattagePerLamp - IfcInteger*
    - *Power used by each lamp in the fixture*
  - *BallastType - IfcBallastTypeEnum*
    - *The type of ballast used in the fixture*

5.6.1.3. RoadMap Issues

Interoperability Value

Disciplines/Applications from which information is needed:

- *Architecture*
- *HVAC*
- *Lighting*

Disciplines/Applications to which information will be supplied:

- *Architecture*
- *HVAC*
- *Lighting*

Value of software supporting this process

In this section, other domain teams will rank the value of software which supports this process, based on IFC. Value from 1-10, 1 being the lowest value, 10 being the highest value.

- *Not assessed for this project*

Sponsor Software Companies

The following organizations that develop software used by AEC end users have shown interest in developing applications which implement the process:

- *Pacific Northwest National Laboratory*
- *Autodesk (was Softdesk)*
- *Visio (was Ketiv)*

5.6.1.4. Issues idendified in reviews

No reviews recorded

Issues

Proposed resolution

- No issues recorded
    - No resolutions recorded

5.7. [CS-2] Code Checking Extensions

The project covers specific application of the code compliance enabling mechanism (R2_CS-1) in serving the disabled access and escape routes code compliance.

The processes are performed by building designers and code enforcement officials during early design and submission stages, respectively.

Codes considered in this project:

- For Access
- For Escape

5.7.1. Codes for Access and Escape

5.7.1.1. Information Analysis by Task

5.7.1.1.1. Task A - Identify Applicable Code Requirements

This step involves selecting or specific code requirements applicable to a particular type of buildings

Input Information:

- *Types, functions and occupancy of building, space and access found on a floor plan*
- *Divisions, sections of and clauses of codes*

Output Information:

- *Specific code requirements to be satisfied*

5.7.1.1.2. Task B - Check for Compliance

This step involves accessing information about building components from the building model, performing additional computations to derived more information not capture in the building model but is essential for the code checking.

Input Information:

- *Building model*
- *Code requirements to be satisfied resulted from previous step.*
- *All derived information necessary for compliance checks.*

Output Information:

- *Compliance checks results*

5.7.1.1.3. Task C - Record Code Violations

This step involves presenting code violation on screen, generating written directions and recording as part of building model for further reference.

Input Information:

- *Compliance checks results*
- *Building model.*

Output Information:

- *Highlight of the violation*
- *Return directions*
- *Building model*

5.7.1.2. IFC Model Impact

This section summarizes the model requirements from all the process tasks analyzed above into two groups Extensions to R1.5.1 model object types and proposed new object types for R2.0.

Extensions to IFC R1.5 object types – Disabled Access

- ***IfcSpace***
  Data
  - *IfcSpaceType – IfcTypeDefinition*
    - *Add to predefined type of IfcSpaceType to cover concept of internal and external with respect to the building envelop.*
- ***IfcZone***
  Data
  - *IfcZoneType*

- *Add to predefined type of IfcZoneType to cover zone with disabled access provision so that checking can be more focus.*

- ***IfcDoor***

  Data
  - *IfcOpenType – IfcTypeDefintion*
    - *Predefined generic types to describe how door is opened (e.g. Sliding, Swing, one-way or two way)*
  - *IfcSwingDirection – IfcTypeDefintion*
    - *Predefined generic types to describe the swing directions with respect to a pivot (pull and push side of the swing to be indicated)*
  - *ClearanceSpace – IfcArea*
    - *Area of a square bounding the swing of a door.*

Extensions to IFC R1.5 object types – Disabled Access

- ***IfcBuilding***

  Data
  - *IfcBuildingUsage – IfcTypeDefinition*
    - *Predefined generic types to describe usage of building e.g. Enum (commercial, residential). This will in turn links to the occupancy load.*
  - *IfcOccupancyLoad – IfcInteger*
    - *Estimated maximum numbers of peoples likely to occupy the building at any one time*

- ***IfcZone***

  Data
  - *IfcZoneUsage – IfcTypeDefinition*
    - *Add to predefined type of IfcZoneUsage to cover fire compartment or zone with fire-protected provision (including smoke, sprinkler) so that checking can be more focus. provision, so that checking can be more focus.*
  - *IfcOccupancyLoad – IfcInteger*
    - *Estimated maximum numbers of peoples likely to occupy the zone at any one time*

- ***IfcStorey***

  Data
  - *IfchasFloorLevel – IfcRea*
    - *Floor Level above ground*
  - *IfchasExitFacility – List [1:N] IfcExitFacility*
    - *List of exit facilities that found in the storey*
  - *IfcOccupancyLoad – IfcInteger*
    - *Estimated maximum numbers of peoples likely to occupy the storey at any one time*

- ***IfcSpace***

  Data
  - *IfcSpaceUsage – IfcTypeDefinition*
    - *Named types to describe usage of a space (e.g. classroom, staircase). This will in turn links to the occupancy load.*
  - *IfcOccupancyLoad – IfcInteger*
    - *Estimate number of people likely to occupy a space at any one time*

New object types required – Disabled Access

- ***IfcRamp***

  Data
  - *IfchasElements – List [1:N] IfcRampElement*
    - *Consists of a list of IfcRampElement*

- ***IfcRampElements***

  Data

- *IfcRampElementsType – IfcTypeDefinition*
  - *Predefined generic types of ramp elements e.g. Enum (flight and landing).*
- *IfchasFootPath – IfcPolyCurve3D*
  - *Path of the center line of the IfcRampElement*
- *IfchasOutline – IfcCurve3D*
  - *A closed 3D curve profile to describe the outline of the IfcRampElement*
- *IfchasSideElements – List [2:N] IfcSideElement*
  - *Side element can be wall, column or balustrade*
- *IfchasEffectiveWidth - IfcReal*
  - *Minimum clear width of a ramp element*
- *IfchasFloorMaterial - IfcTypeDefinition*
  - *Predefined generic types of approved ramp material.*

- ***IfcLanding***

  Data
  - *IfchasEffectiveLength – IfcReal*
    - *Minimum clear length for packing a stationary wheelchair on the landing*
  - *IfchasFloorLevel – IfcReal*
    - *Floor level of a landing*

- ***IfcFlight***

  Data
  - *IfchasVerticalRise – IfcReal*
    - *Change in floor level*
  - *IfchasHorizontalRun – IfcReal*
    - *Length of the run*

- ***IfcSideElement***

  Data
  - *IfcSideElementsType – IfcTypeDefinition*
    - *Predefined generic types of side elements e.g. Enum (wall, column, balustrade).*
  - *IfchasBaluster – IfcBalusterType*
    - *baluster can also be handrails, railing*

- ***IfcBaluster***

  Data
  - *IfcBalusterType - IfcTypeDefinition*
    - *Predefined generic types of baluster or handrail*
  - *IfchasProfile –IfcPolyCurve3D*
    - *Polycurve that defines the path of baluster*
  - *IfchasGrippingArea – IfcReal*
    - *Gripping area of a handrail*

- ***IfcLift***

  Data
  - *IfcLiftType - IfcTypeDefinition*
    - *Predefined types of lift e.g. Enum (Disabled, Cargo, Fire etc)*
  - *IfchasEffectiveWidth – IfcReal*
    - *Minimum clear width for the maneuvering of wheelchair into the lift*
  - *IfchasEffectiveLength – IfcReal*
    - *Minimum clear length for packing a stationary wheelchair in the lift*
  - *IfchasEffectiveTurningArea – IfcArea*
    - *Minimum area for turning of a wheelchair in the life and at the doorway of the lift.*
  - *IfcServingStorey – List [1:N] IfcBuildingStorey*
    - *Stories being served by the Lift*

- ***IfcSymbol***

  Data

- *IfcSymbolType - IfcTypeDefinition*
  - *Predefined types of symbol e.g. Enum (Disabled, Fire etc)*
- *IfcPlacement - IfcPoint*
  - *Position of symbol*

New object types required – Escape Route

- ***IfcExitFacility***

  Data
  - *IfcExitFacilityType – IfcTypeDefinition*
    - *Predefined generic types of exit facilities e.g. Enum (e.g. exit door, exit approach)*
  - *IfchasFireRating – IfcReal*
    - *Number of hours*
  - *IfchasCapacity – IfcInteger*
    - *Maximum numbers of occupant passing through the facility at any one time.*
  - *IfchasEffectiveWidth – IfcReal*
    - *Effective width of the facility*
- ***IfcExitApproach***

  Data
  - *IfcExitApproachType – IfcTypeDefinition*
    - *Predefined generic types of exit approach e.g. Enum (e.g. exit passageway, exit corridor, exit lobby, exist staircase).*
  - *IfcOtherRelatedUsage – List [1:N IfcTypeDefinition*
    - *Predefined generic types of other usage for exit approach e.g. Enum (e.g. area of refuge, area of fire fighting).*
  - *IfchasFireResistanceProvision – List [1:N] IfcFireResistanceProvision*
    - *list of fire resistance provision*
  - *IfchasOpening – List [1:N] IfcOpening*
    - *list of openings*
- ***IfcFireResistanceProvision***

  Data
  - *IfcFireResistanceProvisionType – IfcTypeDefinitoon*
    - *Predefined generic types of fire resistance provision e.g. Enum (e.g. smoke-free, sprinklered, naturally ventilated, mechanically ventilated equipment).*

5.7.1.3. RoadMap Issues

Interoperability issues

Disciplines from which information is needed:

- *Architectural*
- *Structural*
- *HVAC*

▪ *Disciplines for which information is produced:*

- *Architectural*
- *HVAC*
- *Fire Protection*
- *Cost Estimator*
- *Code Enforcement*

Value to AEC Domains

- *Not ascertained*

Sponsor Software Companies

The following organizations that develop software used by AEC end users have shown interest in developing applications which implement the process:

- *National Computer System*

5.7.1.4. Issues Identified in Reviews

No reviews recorded

Issues

Proposed resolution

- No issues recorded
    - No resolutions recorded

5.8. [ES-1] Cost Estimating

5.8.1. Cost Estimating

The intent of cost estimating is to determine the cost of various objects, tasks and resources in the model. A cost estimator will perform several sub-process to get the estimate. First he will determine the purpose and scope of the estimate. He will identify the objects and processes that lie within that scope. He may project costs directly from the type and dimensions of an object, or he may model the materials, tasks and resources to construct or install an object. Once an object, its parts, and its required tasks and resources have been determined, costs may be calculated and applied to the model. This includes the unit costs of resources, the resulting costs of the tasks, the actual cost of manufactured parts, and the summarized costs of the objects.

5.8.1.1. Information Analysis by Task

5.8.1.1.1. Task A - Scope Analysis

The model is analyzed to determine what objects and object information is available for estimating. The purpose of the estimate is also considered.

Input Information:

- *Objects available in the model (spaces, walls, doors, manufactured parts,…)*
- *Type of information in the objects that may be used for estimating. (classifications, material specifications, dimensions…)*
- *Purpose of the estimate (conceptual, detailed, alternate, change order, basis of a bid…)*

output Information:

- *Types of objects that will be used as the basis for the estimate. For example, spaces for a conceptual estimate, or more granular objects like doors and walls for a more detailed estimate.*
- *Identification methods to be used to classify the object in terms of the cost estimating system.*

5.8.1.1.2. Task B - Identify Object

A class of objects is selected for estimating. All instances of that type of object are selected and classified in terms of the cost estimating system. The object selection criterion should be stored so that an estimate of the same scope can be accomplished after changes have been made to the model.

Input Information:

- *Types of objects to be estimated.*
- *The object's class.*
- *The object's specification according to some classification system.*
- *The object's material specification (such as wood, metal, …)*

- *The object's specification requirements (such as fire rating)*
- *The object's dimension attributes.*
- *The 'context' of the object. (for example, the material type of the wall a door installed in)*
- *The object's design status (new, changed, deleted) and version number*
- *Other attributes that may be of use for cost estimating…*

Output Information:

- *Object selection criterion*
- *Aggregation of objects for estimating and/or scheduling*
- *Decomposition objects for estimating and/or scheduling*
- *The classification of the object in terms of the cost estimating system.*

Project Model Usage Requirements:

Existing Classes:

- ***Any object that may impact the cost of the project.***
  Data
  - *dimensional information*
    - *lengths, widths, volumes, …*
  - *specification information*
    - *material, functional specification, structural specification, …*
- ***IfcClassification***
  Data
  - *ClassificationPublisher -> IfcString*
    - *This references the publisher of the cost book or database.*
  - *ClassificationTable -> IfcString*
    - *This references the specific table used.*
  - *ClassificationNotation -> IfcString*
    - *This is the code for the object being classified.*
  - *ClassificationDescription -> IfcString*
    - *This is a readable description of the classification.*

New Classes:

- ***IfcSelectionSpecificaton***
  Data
  - *SelectionSpecification IfcString*
    - *This is a parsable expression that can describe a group of objects in the model based on class, attributes, and relationships to other objects. For example, an estimator may want to save a specification for selecting all wood doors that are installed in drywall partition walls.*
- ***IfcAggregation***
  Data
  - *AggregationElements Set [0:N] Ref IfcProductObject*
    - *This groups together objects that are estimated together.*
- ***IfcConstructionZone***
  Data
  - *ConstructionZone IfcSpaceElement*
    - *An IfcProductObject may be decomposed into several constructions zones. For example, an IfcSlab may be decomposed into several pour zones.*

5.8.1.1.3. Task C - Identify Tasks Needed to Install the Object

The estimator examines the object to determine its construction method. The construction method will specify the tasks that need to be completed to construct the object.

Input Information:

- *Class of the object (wall, door, …)*

- *Attributes of the object (material, finish, …)*
- *Dimensions of the object (height, area…)*

Output Information:

- *IfcWorkGroup - Work objects that group associated tasks.*
- *IfcWorkTask - Tasks required to construct or install the object*
- *IfcResourceObject - Resources required by work tasks..*

Project Model Usage Requirements:

Existing Classes:

- ***IfcWorkGroup***
  Data
  - *WorkGroupTitle IfcString*
    - *This allows several tasks to be grouped together.*
  - *HasParts Set [0:N] ref IfcWorkGroup*
    - *This allows hierarchical groupings.*
  - *ConsistsOf Set [0:N] ref IfcWorkTask*
    - *This allows several tasks to be grouped together.*
- ***IfcWorkTask***
  Data
  - *TaskDescription IfcAttString*
    - *Describes the task*
  - *WorkMeghod IfcAttString*
    - *Describes the work method for the task*
  - *TotalCost IfcCost*
    - *Total cost of the task*
  - *Resources List [0:N] fcResourceObject*
    - *List of resources needed to complete the task*
  - *ResourceQuantity List [0:N] IfcAttReal*
    - *The Quantities of the above resources*
  - *ResourceDuration List [0:N] IfcAttDate*
    - *Time durations for the above resources are needed*
- ***IfcResourceObject***
  Data
  - *ResourceType enum Labor, Equipment, Material*
    - *Specifies the basic type of resource*
  - *ResourceDescription IfcAttString*
    - *Description of the resource. (e.g. Carpenter, Hoist, Forms.)*
  - *HasCost -> IfcUnitCost*
    - *Cost per unit*

New Classes:

- ***No new classes are required for this functionality.***

5.8.1.1.4. Task D - Identify Resources Needed to Install the Object

The estimator determines the resources required to perform each of the tasks.

Input Information:

- *IfcTask*
- *Attributes of the object (material, finish, …)*
- *Dimensions of the object (height, area…)*

Output Information:

- *IfcResource*

Project Model Usage Requirements:

Existing Classes:

- ***IfcResourceObject***

  Data

  - *ResourceType enum Labor, Equipment, Material*
    - *Specifies the basic type of resource*
  - *ResourceDescription IfcAttString*
    - *Description of the resource. (e.g. Carpenter, Hoist, Forms.)*
  - *HasCost -> IfcUnitCost*
    - *Cost per unit*

New Classes:

- ***No new classes are required for this functionality.***

5.8.1.1.5. Task E - Determine Unit Costs

Once quantities have been determined for the 'overall' object or its tasks and resources, unit costs are applied. If cost is being modeled using tasks and resources, unit costs are selected for the resources. If cost is being modeled based on a unit cost for the overall object, a unit cost is select for the overall object.

Input Information:

· *Object to be costed*
· *Resources to be costed*
· *Unit costs (possibly from an estimating system or price book)*

Output Information:

· *IfcUnitCost*

Project Model Usage Requirements:

Existing Classes:

- ***IfcUnitCost***

New Classes:

- ***No new classes are needed.***

5.8.1.1.6. Task F - Calculate Costs

The object and resource quantities and the selected unit costs are used to calculate the cost of the object or its resources.

Input Information:

· *Object's 'overall' quantity*
· *Resource quantities*
· *Unit costs*

Output Information:

· *Resource cost (if tasks and resources are used to model the cost)*
· *Object cost (if the cost is based on the object's 'overall' quantity)*

Project Model Usage Requirements:

Existing Classes:

- ***IfcCost***
- ***IfcUnitCost***

New Classes:

- ***No new classes are needed.***

5.8.1.1.7. Task G - Summarize Costs

If an object's cost is based on the costs of its tasks and resources or on the costs of its component parts, summarize these costs at the task and object level.

You must be able to place a cost on any object in the model, including aggregations and decompositions. Furthermore, objects will need multiple costs. For instance, you may need to store original budget cost for the object, final estimated cost, and actual installed cost for one object.

To organize costs in a format that is meaningful, they are compiled in Cost Schedules. The cost elements in a schedule and the physical objects whose cost they represent, should have references to each other. That is, from the physical object, you should be able to find all cost schedule elements that describe the cost of the object. And from the cost schedule element, you should be able to find the physical object that it references.

Input Information:

- *Resource costs*
- *Costs of component parts*

Output Information:

- *Task costs*
- *Object cost*
- *Cost schedule*

Project Model Usage Requirements:

Existing Classes:

- ***IfcWorkTask***
  Data
  - TaskCost IfcCost
  - Total cost of the task

- ***IfcProductObject***
  Data
  - ProductCost IfcCost
  - Cost impact of the product object.

- ***IfcCostScheduleElement***
  Data
  - TotalCost IfcCost
  - Total cost of this entry in the cost schedule
  - UnitCost IfcUnitCost
  - Cost per unit for this entry in the cost schedule
  - Description IfcAttString
  - Description for this entry in the cost schedule
  - Quantity IfcAttReal
  - Quantity for this entry in the cost schedule (in terms of UnitCost unit)

- ***IfcProjectObject***
  Data
  - Costs LIST[0:N] IfcCost
  - CostSchedules LIST[0:N] IfcCostSchedule
  - CostScheduleElements LIST[0:N] IfcCostScheduleElement

New Classes:

- ***No new classes are needed.***

5.8.1.2. IFC Model Impact

This section summarizes the model requirements from all the process tasks analyzed above into two groups Extensions to R1.5.1 model object types and proposed new object types for R2.0.

Usage/Extensions to R1.5.1 object types

- ***IfcProductObject***
  Data
  - *ConstructionZones Set [0:n] ref IfcConstructionZone*
    - *The decomposition of a product object into construction zones. For example, an IfcSlab may be decomposed into several pour zones.*
- ***IfcWorkTask***
  Data
  - *We need to make sure that the current IfcWorkTask is able to specify any number of resource usages.*
  - *Resources List [0:N] fcResourceObject*
    - *List of resources needed to complete the task*
  - *ResourceQuantity List [0:N] IfcAttReal*
    - *The Quantities of the above resources*
  - *ResourceDuration List [0:N] IfcAttDate*
    - *Time durations for the above resources are needed*
- ***IfcProjectObject***
  Data
  - *Costs List[0:?] IfcCost*
    - *We should allow for more than one cost on a project object, since it may have different costs, depending on your viewpoint or the stage of design. For example, is a cost an actual cost or an estimated cost?*
  - *CostSchedules List[0:?] IfcCostScheduleObject*
    - *The IfcProjectObject should be able to reference cost schedules that explain the cost breakdown of the object. This allows an object's costs to be reported in any number of formats.*
  - *CostScheduleElements List[0:?] IfcCostScheduleElements*
    - *The IfcProjectObject should be able to reference cost schedules that display the object's cost in the context of the cost estimate.*
- ***IfcCost***
  Data
  - *CostType Enumeration???*
    - *In the current model (1.5), ProductCost is used to represent the cost of an IfcProductObject. But as mentioned above, the meaning of a cost is not precisely defined. For instance, is a ProductCost the sum of the cost of its sub-object and work objects? Is it a manufacturer's cost, or does it include cost of installation?*
    - *We should specify Cost Types that would help define the meaning of an IfcCost. Some possible cost types are:*
      - *Singular cost - Costs calculated based on a unit cost and a quantity.*
      - *Aggregate cost - Sum of the costs of an object's parts.*
      - *Target cost (allowable cost?) - Used to communicate the allowable expenditure for an object. This may also be a range.*

New object types required

- ***IfcSelectionSpecificaton***
  Data
  - *SelectionSpecification IfcString*

- *A parsable expression that identifies a group of objects in the model based on class, attributes, and relationships to other objects. Example: Select all wood doors that are installed in drywall partition walls in a particular construction zone.*

- ***IfcAggregation***
  Data
  - *AggregationElements   Set [0:N] Ref IfcProductObject*
    - *This groups together objects that are estimated and/or scheduled together*

- ***IfcConstructionZone***
  Data
  - *ConstructionZone   IfcSpaceElement*
    - *A project may have multiple "construction zones" which are defined by the construction planner, for scheduling tasks and assigning resources and methods. For instance, a construction planner may sequence foundation pours starting at zone 1 and proceeding through the last zone scheduled.*

5.8.1.3. RoadMap Issues

Interoperability Issues

Disciplines/Applications from which information is needed:

- *Architecture*
- *Other disciplines that provide attribute sets to describe objects.*

Disciplines/Applications to which information will be supplied:

- *Architecture*
- *Estimating*
- *Scheduling*
- *Facilities Management*
- *Other disciplines that provide attribute sets to identify objects*

Value of software supporting this process

- *Scheduling - 1*
- *Estimating - 1*
- *Facilities Management - 3*

Sponsor Software Companies

The following organizations that develop software used by AEC end users have shown interest in developing applications which implement the process:

- *Timberline Software*

5.8.1.4. Issues identified in reviews

No reviews recorded

Issues

Proposed resolution

- No issues recorded
  - No resolutions recorded

5.9. [FM-3] Property Management *(Building Owner's viewpoint)*

5.9.1. Grouping IFC objects

This process can be performed at any stage in the lifecycle of the building, but it has been designed for the Property manager. Groups can consist of IFC Object, and the user can use the Group object to make links to the users own private objects.

The need for grouping can be caused by any management purpose, like new department, workgroup, cleaning area, renovation, fire zone etc. In this process the property manager can create new groups from selected objects. These groups can be used for any administrative or management purposes. All material or quantitative information is calculated from the IFC model. The model information can be used together with owner's own or other external database information to evaluate operational costs or other needed values.

5.9.1.1. Information Analysis by Task

There are three processes in the Grouping IFC Objects. Manage requirements is properly supported by a Facility Management Software tool, while Grouping IFC is a IFC based application. Evaluation and other functions can be based on the users own software, it must just be able to read the needed IFC Objects.

The first task is to define the grouping purpose, which defines the classification of the group. Then the objects for new group can be selected through various methods:

- any objects selected by the user
- filtered objects (type, properties or other selection key) selected by the user
- filtered objects in the whole model

After the selection is completed the user can give a description to the group.

5.9.1.1.1. Task A - Management requirements

The user makes decisions about what he would like group.

Input Information:

- *Any needs for grouping*

Output Information:

- *Selection criteria*
- *Space type either/or*
- *Floor type*
- *Department use of spaces*
- *Cleaning*
- *etc.*

Project Model Usage Requirements:

Existing Classes:

- ***none for this task***

New Classes:

- ***none for this task***

5.9.1.1.2. Task B - Grouping

The Grouping process reads information about what should be grouped. A number of object is selected from the IFC model either by IFC Objects type/attribute or by the user picks a number of objects from the model. A new grouping object is made and the identification of the selected objects a store/linked to the grouping object.

Input Information:

- *Section criteria*
- *IFC Objects (IFC Space, IFC Layered Element etc.)*

Output Information:

- *IFC Grouping object*
- *Identification*
- *Description*
- *Classification of groups*

Project Model Usage Requirements:

Existing Classes:

- ***IFC Object***
  Data
  - *Identification*
  - *Attribute already in IFC objects*

New Classes:

- ***IFC Grouping Object***
  Data
  - *Identification*
  - *Description*
  - *Classification of group from a project specific list*

5.9.1.1.3. Task C - Management, rental etc.

Use IFC Grouping object to make connection to private database systems or to makes reports and drawings.

Input Information:

- *IFC Grouping Object*
- *Own data: Cost, taxes etc.*

Output Information:

- *Evaluation result (Rental, drawings)*

Project Model Usage Requirements:

Existing Classes:

- ***IFC Objects***
  Data
  - *Identification*
  - *Attribute already in IFC objects*

New Classes:

- ***{{ Object type name }}***
  Data
  - *{{ Data description type }}*
    - *{{ notes }}*

5.9.1.2. IFC Model Impact

This section summarizes the model requirements from all the process tasks analyzed above into two groups Extensions to R1.5.1 model object types and proposed new object types for R2.0.

Usage/Extensions to R1.5.1 object types

- ***IFC Space***

Data
- Room number (new)
- Name of space (new)
- Equipment (Electrical and HVAC) list in a string (new)

- ***IFC DoorType***
  Data
  - Thermal Rating (new)
  - Security Rating (new)
  - Change door hardware type to hardware type

- ***IFC Windowtype***
  Data
  - Thermal Rating (new)
  - Numbers of glasses (new)
  - Fire Rating for the window (new)
  - Acoustic Rating (new)
  - Security Rating (new)
  - Pointer to hardware type (new)
  - External/Internal (new ?)

- ***IFC LayeredElement***
  Data
  - Thermal Rating (new)
  - Fire Rating for the window (new)
  - Acoustic Rating (new)
  - External/Internal (new ?)

- ***IFC WallType***
  Data
  - Measure areal of an external wall ?
  - Remove Thermal Rating
  - Remove Fire Rating
  - Remove Acoustic Rating

New object types required

- ***IFC Grouping Object***
  Data
  - Identification
  - Description
  - Classification of group from a project specific list

5.9.1.3. RoadMap Issues

Interoperability Issues

Applications from which information is needed:

- *Architectural (Spaces, Wall, Floors, Doors, Windows)*
- *Others in the future*

Applications for which information is produced:

- *Facility Management*
- *Any program that needs groups of objects*

Value of software supporting this process

- *{{In this section, please allow for the other domains to rank your process in order of precedance for their domain, this allows us to examine the issue on a group as well as an individual level}}*
- *{{ discipline 1 }} - {{value from 1-10, 1 being the highest value, 10 being the lowest]]*
- *{{ discipline 2 }} - {{value from 1-10]]*

Sponsor Software Companies

The following organizations that develop software used by AEC end users have shown interest in developing applications which implement the process:

- *Viatek FM*
- *Visio Corporation*
- *AIO group (Finland)*

5.9.1.4. Issues identified in reviews

No reviews recorded

Issues

Proposed resolution

- No issues recorded
  - No resolutions recorded

5.9.2. Linking the maintenance objects to the IFC objects

The different building elements in the building are linked to a maintenance object. The guarantees, maintenance periods and maintenance history of these elements is stored in the maintenance object. The property manager can check from this information when maintenance operations should be done and if all necessary operations are made according to the guarantee terms. The grouping mechanism is identical to the groping activities

5.9.2.1. Information Analysis by Task

There are three processes in the Linking the maintenance objects to the IFC objects: Management requirements, Maintenance linking and Maintenance information handling.

First task is to define the selection criteria for a maintenance group. Then the objects for new group can be selected through various methods:

- any objects selected by the user
- filtered objects (type, properties or other selection key) selected by the user
- filtered objects in the whole model

After the selection is completed the user can give a description to the new maintenance group.

If the selected objects already belong to some maintenance group, the application should warn the user about it and ask for instructions for further operations.

When the maintenance groups are formed the user can use those as the selection criteria for different maintenance operations and reports. All maintenance data is stored in the maintenance object and the IFC object data should be available from the actual objects. The first task is to define the grouping purpose, which defines the classification of this group. Then the objects for new group can be selected through various methods:

- any objects selected by the user
- filtered objects (type, properties or other selection key) selected by the user
- filtered objects in the whole model

After the selection is completed the user can give a description to the group.

5.9.2.1.1. Task A - Management requirements

The user makes decisions about what he would like maintain.

Input Information:

· *Any needs for maintenance*

Output Information:

· *Selection criteria*

Project Model Usage Requirements:

Existing Classes:

- ***none for this task***

New Classes:

- ***none for this task***

5.9.2.1.2. Task B - Maintenance linking

IFC objects are linked to a Maintenance object. The IFC Objects are selected according to the selection criteria. A new Maintenance object is defined.

Input Information:

· *Section criteria*
· *IFC Objects (IFC Space, IFC Layered Element etc.)*

Output Information:

· *IFC Maintenance object*

Project Model Usage Requirements:

Existing Classes:

- ***IFC Object***
  Data
  - *ID*

New Classes:

- ***IFC Maintenance Object***
  Data
  - *Identification*
  - *Description*
  - *Classification*
  - *Delivery date*
  - *Guarantee terms (Pointer to)*
  - *Guarantee ending date*
  - *Maintenance period*
  - *Last maintenance date*
  - *Maintenance handling*
  - *Maintenance Instruction (Pointer to)*
  - *Maintenance history (Pointer to)*
  - *Inspection intervals*
  - *Condition report*
  - *Last inspection date*
  - *Inspection handling*
  - *Inspection history (Pointer to)*

- *Priority*
- *Cost*

5.9.2.1.3. Task C - Maintenance information handling

Use Maintenance Object to produce Maintenance operation schedules and instructions. Maintenance instruction are stored outside the IFC model.

Input Information:

- *IFC Maintenance Object*
- *External databases/links to maintenance instructions.*

Output Information:

- *Maintenance operations*

Project Model Usage Requirements:

Existing Classes:

- ***IfcCost***

5.9.2.2. IFC Model Impact

This section summarizes the model requirements from all the process tasks analyzed above into two groups Extensions to R1.5.1 model object types and proposed new object types for R2.0.

Extensions to IFC R1.5 object types

- ***IFC Objects (IFCWindow)***

  Data
  - *Id*
  - *Type (IFCWindowtype)*

New object types required in IFC R2.0

- ***IFC Maintenance Object***

  Data
  - *Identification*
  - *Description*
  - *Classification*
  - *Delivery date*
  - *Guarantee terms (Pointer to)*
  - *Guarantee ending date*
  - *Maintenance period*
  - *Last maintenance date*
  - *Maintenance handling*
  - *Maintenance Instruction (Pointer to)*
  - *Maintenance history (Pointer to)*
  - *Inspection intervals*
  - *Last inspection date*
  - *Inspection handling*
  - *State of condition*
  - *Inspection history (Pointer to)*
  - *Priority*
  - *Cost*

5.9.2.3. RoadMap Issues

Interoperability Issues

Disciplines/Applications from which information is needed:

- *Architectural (Wall, Doors, Windows, Floors)*
- *HVAC*

Disciplines/Applications to which information will be supplied:

- *Facility Management (Maintenance)*

Value of software supporting this process

- *None provided*

Sponsor Software Companies

The following organizations that develop software used by AEC end users have shown interest in developing applications which implement the process:

- *Viatek FM*
- *Visio Corporation*

5.9.2.4. Issues identified in reviews

No reviews recorded

Issues

Proposed resolution

- No issues recorded
    - No resolutions recorded

5.10. [FM-4] Occupancy Planning

5.10.1. Occupancy Planning

The occupancy planner (includes interior designers, facilities managers, architects, furniture dealers, etc.) applies standards during the assignment of people and organizations to interior spaces. This process occurs during the initial planning of space occupancy, and whenever that occupancy needs to change (company reorganization, company growth, etc.)

5.10.1.1. Information Analysis by Task

5.10.1.1.1. Task A - Assess Move Request

Assess request with respect to occupant information, company policies, regulatory requirements. Identify FF&E required for the occupant, and generate space programme.

Input Information:

- *Request for Move*
    - *Space area requirements*
    - *Space service requirements (cooling requirements, gases required, etc.)*
    - *Adjacencies/affinities relationships (location)*
    - *FF&E required*

  - *Department requirements*
  - *Occupant list*
  - *Target Occupancy Date*
  - *Budget*
  - *Environmental requirements, e.g. daylight, purification of air*
  - *Special requirements (e.g. raised floor)*
- *Company policies and standards*
  - *Relationship between occupant position/title/department and space standard*
  - *Relationship between occupant position/title/department and equipment standard*
  - *Relationship between occupant position/title/department and furnishings standard*
- *Regulatory requirements*
  - *National or local facilities regulations (e.g. ADA, OSHA)*
  - *Local Fire/Electrical Codes*

Output Information:

- *Space programme*
  - *Number and sizes of spaces*
  - *Requested Space locations*
  - *Space characteristics*
  - *Occupant list*
  - *FF&E lists (existing/new)*
  - *Target Occupancy Date*
  - *Budget*
  - *Special requirements*

Project Model Usage Requirements:

Existing Classes:

- ***IfcSpaceProgramme (referenced by IfcProgrammeGroup)***
  - *{{ all attributes described in version 1.5 spec + the following }}*
  - *RequestedLocation: Ref. to IfcProductObject*
  - *<this can reference a building, storey, or space>*
  - *ServiceRequirements: Set [0:N] of IfcString*
  - *RequiredFF&E: Set [0:N] of Ref. to IfcTypeDefinition*
  - *TargetDate: IfcDate*
  - *Budget: IfcCost*
  - *SpecialRequirements: Set [0:N] of IfcSpaceRequirement*
- ***IfcActor (referenced in IfcOccupancySchedule, IfcMoveAction, IfcPlan)***
- ***IfcControlObject (supertype of IfcPlan, IfcOccupancySchedule, IfcMoveActionConstraint)***
- ***IfcCost (used in IfcPlan)***
- ***IfcDate (used in Att_ScheduleData, IfcMoveActionConstraint, IfcPlan)***
- ***IfcElement (referenced in IfcMoveAction)***
- ***IfcPlan (supertype of IfcMovePlan)***
- ***IfcProcessObject (used in IfcOccupancySchedule, supertype of IfcMoveAction)***
- ***IfcProgrammeGroup (referenced in IfcMovePlan)***
- ***IfcProject (referenced in IfcPlan)***
- ***IfcRelSequence (referenced in IfcOccupancySchedule)***
- ***IfcSpace (referenced in IfcMoveAction)***
- ***IfcString (used in IfcPlan, IfcMoveActionConstraint)***
- ***IfcTimeDuration (used in Att_ScheduleData)***
- ***IfcWorktask (referenced in IfcMovePlan)***

New classes:

- ***IfcPlan   subtype of IfcControlObject, supertype of IfcMovePlan***
  - *PlanID: IfcString*
  - *PlanName: IfcString*
  - *PlanDescription: Set [0:N] of IfcString*
  - *Project: Ref. to IfcProject*
  - *PlanCreators: Set [0:N] of IfcActor*
  - *CreationDate: IfcDate*
  - *Approval: Ref. to IfcApproval*
  - *Budget: IfcCost*
- ***Att_ScheduleData   Extension Attributeset for any object that uses schedule data set***
  - *TotalDuration: IfcTimeDuration*
  - *ScheduledStartDate: IfcDate*
  - *ScheduledFinishDate: IfcDate*
  - *ActualStartDate: IfcDate*
  - *ActualFinishDate: IfcDate*
  - *EarlyStartDate: IfcDate*
  - *EarlyFinishDate: IfcDate*
  - *LateStartDate: IfcDate*
  - *LateFinishDate: IfcDate*
  - *TotalFloat: IfcTimeDuration*
  - *DaysRemaining: IfcTimeDuration*
- ***IfcOccupancySchedule   subtype IfcControlObject***
  - *OccupyingActions: Set [0:N] of Ref. to IfcProcessObject*
  - *<this contains a set of IfcWorktask and IfcMoveAction>*
  - *PredAndSuccs: Set [0:N] of Ref. to IfcRelSequence*
  - *Schedule: Att_ScheduleData*
  - *Responsible: Ref. to IfcActor*
- ***IfcMoveAction   subtype of IfcProcessObject***
  - *OccupantsToMove: Set [0:N] of Ref. to IfcActor*
  - *FF&EToMove: Set [0:N] of Ref. to IfcElement*
    - *<these can be IfcFurniture, IfcEquipment, IfcSystemFurniture, etc.>*
  - *MoveFrom: Ref. to IfcSpace*
  - *MoveTo: Ref. to IfcSpace*
  - *Schedule: Att_ScheduleData*
  - *Constraints: Set [0:N] of IfcMoveActionConstraint*
  - *Responsible: Ref. to IfcActor*
- ***IfcMoveActionConstraint   subtype of IfcControlObject***
  - *ConstraintType: IfcString*
  - *<e.g. must be out by, etc.>*
  - *ConstraintDate: IfcDate*
- ***IfcMovePlan   subtype of IfcPlan***
  - *OccupancySchedule: Ref. to IfcOccupancySchedule*
  - *RequiredWork: Set [0:N] of Ref. to IfcWorktask*
  - *ProgramGroupToBeMoved: Ref. to IfcProgrammeGroup*
  - *<this programme group references a set of Space Programs>*

5.10.1.1.2. Task B - Evaluate Candidate Solutions

Compare space programme to available (incl. existing or added) spaces to find candidate solutions including the changes of spaces and FF&E.

Input Information:

- *Space programme*
- *Space inventory*
- *List of candidate spaces and characteristics (see Space Programme)*

Output Information (assuming candidate space exists):

- *Schematic Design*
  - *space assignment*
  - *schematic drawings*
- *Required changes*
  - *Space changes*
  - *FF&E changes*

Project Model Usage Requirements:

Existing Classes:

- ***IfcActor (referenced in IfcInventory)***
- ***IfcArea (used in IfcSpaceInventory)***
- ***IfcDate (used in IfcInventory)***
- ***IfcInteger (used in IfcSpaceInventory)***
- ***IfcProductObject (referenced in IfcInventory)***
- ***IfcSpace (referenced by IfcSpaceInventory)***
- ***IfcSpaceElement (referenced in IfcSpaceInventory)***
- ***IfcString (used in IfcMovePlan, IfcSpaceInventory, IfcInventory)***

New Classes:

- ***IfcMovePlan extended from last step***
  - *{{ all items described in previous process steps + the following }}*
    - *Documents: Set [0:N] of Ref. to IfcDocument (to be defined in the next process)*
    - *<e.g. schematic drawings, etc.>*
- ***IfcInventory <Abstract class> supertype of IfcSpaceInventory, IfcFurnitureInventory, and IfcEquipmentInventory***
  - *InventoryDescription: IfcString*
  - *InventoryScope: Ref. to IfcProductObject {{this can reference to a building, storey, or space}}*
  - *InventoryJurisdiction: Ref. to IfcActor*
  - *InventoryResponsible: Set [0:N] of Ref. to IfcActor*
  - *LastUpdateDate: IfcDate*
- ***IfcSpaceInventory subtype of IfcInventory***
  - *HasSpaces: Set [0:N] of Ref. to IfcSpaceElement {{this allows to include zones}}*
  - *TotalSpaces: IfcInteger*
  - *TotalNetArea: IfcArea*

1.1.1.1.1 Task C - Proposed Move Plan

During the design and generation of drawings, we allow for client review and approval. Define temporary staging areas, generate schedules, identify sources of all FF&E required and generate a cost estimate.

Input Information:

- *Schematic design*
- *Required changes*

Output Information:

- *Proposed move plan*
  - *Drawing*

- *Schedule*
- *Cost estimate*

Project Model Usage Requirements:

Existing Classes:

- ***IfcCostSchedule (used in IfcMovePlan)***

New Classes:

- ***IfcMovePlan extended from last step***
  - *{{ all items described in previous process steps + the following }}*
  - *ProjectCostEstimate: IfcCostSchedule*

5.10.1.1.3. Task D - Approval Process

Occupant and management review proposed move plan and either approve (possibly with constraints) or rejects --> revert to previous steps.

Input Information:

- *Proposed move plan*
  - *Drawing*
  - *Schedule*
  - *Cost estimate*

Output Information:

- *Approval constraints*
  - *Limitations on move plan*

Project Model Usage Requirements:

Existing Classes:

- ***None in this step***

New Classes:

- ***IfcMovePlan***

1.1.1.1.2 Task E - Complete Move Plan

Modify proposed plan to comply with constraints. Generate work orders and purchase orders.

Input Information:

- *Proposed move plan*
- *Approval constraints*

Output Information:

- *Approved plan*
- *record drawing set*
- *move schedule*
- *installation schedule*
- *work orders*
- *purchase orders*

Project Model Usage Requirements:

Existing Classes:

- ***IfcActor (referenced in Att_PurchaseOrder, Att_WorkOrder)***
- ***IfcCost (used in Att_PurchaseOrder, Att_PurchaseOrderItem)***
- ***IfcCostSchedule (used in Att_WorkOrder)***

- ***IfcDate (referenced in Att_PurchaseOrder, Att_WorkOrder)***
- ***IfcInteger (used in Att_PurchaseOrder, At_PurchaseOrderItem)***
- ***IfcProductObject (referenced in Att_WorkOrder)***
- ***IfcReal (used in Att_PurchaseOrderItem)***
- ***IfcString (used in Att_PurchaseOrder, Att_PurchaseOrder, Att_PurchaseOrderItem, Att_WorkOrder)***
- ***IfcWorkTask (referenced in Att_WorkOrder)***
- ***IfcUnit (used in Att_PurchaseOrderItem)***

New Classes:

- ***IfcMovePlan extended from last step***
  - *{{ all items described in previous process steps + the following }}*
  - *WorkOrders: Set [0:N] of Ref. to IfcDocument*
  - *{{ List of references to work orders necessary to complete the Occupancy Schedule}}*
  - *PurchaseOrders: Set [0:N] of Ref. of IfcDocument*
  - *{{ List of references to purchase orders necessary to complete the Occupancy Schedule}}*
- ***WorkOrder of IfcTypeDefinition of DocumentType in IfcDocument***
  - *target object = “IfcDocument”*
  - *shared = Att_DocumentType*
  - *occurance = Att_WorkOrder*
- ***Att_WorkOrder***
  - *TransactionCode: IfcString*
  - *RequestID: IfcString*
  - *Facility: Set [1:N] of Ref. to IfcProductObject*
  - *DateOfRequest: IfcDate*
  - *ShortJobDescription: IfcString*
  - *JobDescription: Set [0:N] of IfcString*
  - *Justification: Set [0:N] of IfcString*
  - *IfNotAccomplished: Set [0:N] of IfcString*
  - *WorkRequest: Ref. to IfcWorkTask*
  - *EstimatedCost: IfcCostSchedule*
  - *ContractType: Enum (InHouse, SelfHelp, Contract)*
  - *Budget: Ref. to IfcBudget*
  - *RequestBy: Ref. to IfcActor*
  - *RequestTo: Ref. to IfcActor*
  - *AdditionalContact: Ref. to IfcActor*
  - *Approval: Ref. to IfcApproval*
- ***PurchaseOrder of IfcTypeDefinition of DocumentType in IfcDocument***
  - *target object = “IfcDocument”*
  - *shared = Att_DocumentType*
  - *occurance = Att_PurchaseOrder*
- ***Att_PurchaseOrder***
  - *PurchaseOrderNo: IfcString*
  - *CompanyTitle: Ref. to IfcActor*
  - *SupplierName: Ref. to IfcActor*
  - *Date: IfcDate*
  - *Remark: Set [0:N] of IfcString*
  - *DateRequired: IfcString*
  - *DateScheduled: IfcDate*
  - *DateActual: IfcDate*
  - *FOB: IfcBoolean*

  - *ShipMethod: IfcString*
  - *TotalCost: IfcCost*
  - *TotalItems: IfcInteger*
  - *PurchaseItems: List [0:N] of Att_PurchaseOrderItem*
  - *Approval: Ref. to IfcApproval*

- ***Att_PurchaseOrderItem***
  - *ItemNumber: IfcInteger*
  - *Quantity: IfcReal*
  - *Code: IfcString*
  - *Unit: IfcUnit*
  - *UnitPrice: IfcCost*
  - *TotalCost: IfcCost*
  - *InvoiceAmount: IfcCost*
  - *TotalBalance: IfcCost*
  - *InPurchaseOrder: Ref. to IfcDocument*

1.1.1.1.3 Task F - Implement Move Plan

Purchase FF&E. Perform work orders. Deal with change orders. Complete staging space. Move the occupant.

Input Information:

- *Approved plan*

Output Information:

- *As-built change <change notes for drawings and documents>*

Project Model Usage Requirements:

Existing Classes:

- ***IfcActor (referenced by Att_ChangeOrder)***
- ***IfcDate (used by Att_ChangeOrder)***
- ***IfcString (used by Att_ChangeOrder)***

New Classes:

- ***IfcMovePlan extended from last step***
  - *{{ all items described in previous process steps + the following }}*
  - *ChangeOrders: Set [0:N] of Ref. to IfcDocument*
  - *<set of references to change orders to accomplish adjustments to the Occupancy Schedule>*
- ***ChangeOrder of IfcTypeDefinition of DocumentType in IfcDocument***

  target object = "IfcDocument"
  shared = Att_DocumentType
  occurance = Att_ChangeOrder

- ***Att_ChangeOrder***
  - *ChangeOrderNo: IfcString*
  - *Description: Set [0:N] of IfcString*
  - *Date: IfcDate*
  - *IssuedBy: Ref. to IfcActor*
  - *IssuedTo: Ref. to IfcActor*
  - *Approval: Ref. to IfcApproval*

1.1.1.1.4 Task G - Updates

Revised documentation and databases to reflect new and revised spaces and assets.

Input Information:

- *Approved plan – as modified through the implementation*
- *As-built changs*
- *Space/FF&E inventory*

Output Information:

- *As-built drawings /updated FM models*
- *updated space/FF&E inventory*

Project Model Usage Requirements:

Existing Classes:

- ***IfcCost (used in IfcFurnitureInventory, IfcEquipmentInventory)***
- ***IfcElement (used in IfcFurnitureInventory)***
- ***IfcEquipment (used in IfcEquipmentInventory)***

New Classes:

- ***IfcFurnitureInventory subtype of IfcInventory***
  - *TotolValueOriginal: IfcCost*
  - *TotalValue: IfcCost*
  - *FurnitureInventory: Set [0:N] of Ref. to IfcElement*
    - *<contains set of IfcFurniture and IfcWorkstation>*
- ***IfcEquipmentInventory subtype of IfcInventory***
  - *TotolValueOriginal: IfcCost*
  - *TotalValue: IfcCost*
  - *EquipmentInventory: Set [0:N] of Ref. to IfcEquipment*

5.10.1.2. IFC Model Impact

Usage/Extensions to R1.0 object types

- ***IfcActor (referenced by Att_ChangeOrder, Att_PurchaseOrder, Att_WorkOrder, IfcInventory, IfcOccupancySchedule, IfcMoveAction, IfcPlan)***
- ***IfcArea (used in IfcSpaceInventory)***
- ***IfcControlObject (supertype of IfcPlan, IfcOccupancySchedule, IfcMoveActionConstraint)***
- ***IfcCost (used in Att_PurchaseOrder, Att_PurchaseOrderItem, IfcFurnitureInventory, IfcEquipmentInventory, IfcPlan)***
- ***IfcCostSchedule (used in IfcMovePlan, Att_WorkOrder)***
- ***IfcDate (referenced in Att_PurchaseOrder, Att_WorkOrder, Att_ChangeOrder, Att_ScheduleData, IfcMoveActionConstraint, IfcPlan, IfcInventory)***
- ***IfcElement (referenced in IfcMoveAction, IfcFurnitureInventory)***
- ***IfcEquipment (used in IfcEquipmentInventory)***
- ***IfcInteger (used in Att_PurchaseOrder, At_PurchaseOrderItem, IfcSpaceInventory)***
- ***IfcPlan (supertype of IfcMovePlan)***
- ***IfcProcessObject (used in IfcOccupancySchedule, supertype of IfcMoveAction)***
- ***IfcProductObject (referenced in Att_WorkOrder, IfcInventory)***
- ***IfcProgrammeGroup (referenced in IfcMovePlan)***
- ***IfcProject (referenced in IfcPlan)***
- ***IfcReal (used in Att_PurchaseOrderItem)***

- ***IfcRelSequence (referenced in IfcOccupancySchedule)***
- ***IfcSpace (referenced by IfcSpaceInventory, IfcMoveAction)***
- ***IfcSpaceElement (referenced in IfcSpaceInventory)***
- ***IfcSpaceProgramme (referenced by IfcProgrammeGroup)***
  - *{{ all attributes described in version 1.0 spec + the following }}*
  - *RequestedLocation: Ref. to IfcProductObject*
  - *<this can reference a building, storey, or space>*
  - *ServiceRequirements: Set [0:N] of IfcString*
  - *RequiredFF&E: Set [0:N] of Ref. to IfcTypeDefinition*
  - *TargetDate: IfcDate*
  - *Budget: IfcCost*
  - *SpecialRequirements: Set [0:N] of IfcSpaceRequirement*
- ***IfcString (used in Att_ChangeOrder ,Att_PurchaseOrder, Att_PurchaseOrder, Att_PurchaseOrderItem, Att_WorkOrder, IfcMovePlan, IfcSpaceInventory, IfcInventory, IfcPlan, IfcMoveActionConstraint)***
- ***IfcTimeDuration (used in Att_ScheduleData)***
- ***IfcUnit (used in Att_PurchaseOrderItem)***
- ***IfcWorkTask (referenced in Att_WorkOrder, IfcMovePlan)***

New object types required

- ***IfcPlan subtype of IfcControlObject, supertype of IfcMovePlan***
  - *PlanID: IfcString*
  - *PlanName: IfcString*
  - *PlanDescription: Set [0:N] of IfcString*
  - *Project: Ref. to IfcProject*
  - *PlanCreators: Set [0:N] of IfcActor*
  - *CreationDate: IfcDate*
  - *Approval: Ref. to IfcApproval*
  - *Budget: IfcCost*
- ***Att_ScheduleData Extension Attributeset for any object that uses schedule data set***
  - *TotalDuration: IfcTimeDuration*
  - *ScheduledStartDate: IfcDate*
  - *ScheduledFinishDate: IfcDate*
  - *ActualStartDate: IfcDate*
  - *ActualFinishDate: IfcDate*
  - *EarlyStartDate: IfcDate*
  - *EarlyFinishDate: IfcDate*
  - *LateStartDate: IfcDate*
  - *LateFinishDate: IfcDate*
  - *TotalFloat: IfcTimeDuration*
  - *DaysRemaining: IfcTimeDuration*
- ***IfcOccupancySchedule subtype IfcControlObject***
  - *OccupyingActions: Set [0:N] of Ref. to IfcProcessObject*
  - *PredAndSuccs: Set [0:N] of Ref. to IfcRelSequence*
  - *Schedule: Att_ScheduleData*
  - *Responsible: Ref. to IfcActor*
- ***IfcMoveAction subtype of IfcProcessObject***
  - *OccupantsToMove: Set [0:N] of Ref. to IfcActor*
  - *FF&EToMove: Set [0:N] of Ref. to IfcElement*
  - *MoveFrom: Ref. to IfcSpace*

- *MoveTo: Ref. to IfcSpace*
- *Schedule: Att_ScheduleData*
- *Constraints: Set [0:N] of IfcMoveActionConstraint*
- *Responsible: Ref. to IfcActor*
- *OccupancySchedule: Ref. to IfcOccupancySchedule*
- *RequiredWork: Set [0:N] of Ref. to IfcWorktask*
- *ProgramGroupToBeMoved: Ref. to IfcProgrammeGroup*
- *Documents: Set [0:N] of Ref. to IfcDocument*
- *ProjectCostEstimate: IfcCostSchedule*
- *WorkOrders: Set [0:N] of Ref. to IfcDocument*
- *PurchaseOrders: Set [0:N] of Ref. of IfcDocument*
- *ChangeOrders: Set [0:N] of Ref. to IfcDocument*

- ***IfcMoveActionConstraint subtype of IfcControlObject***
  - *ConstraintType: IfcString*
  - *ConstraintDate: IfcDate*
- ***IfcInventory <Abstract class> supertype of IfcSpaceInventory, IfcFurnitureInventory, and IfcEquipmentInventory***
  - *InventoryDescription: IfcString*
  - *InventoryScope: Ref. to IfcProductObject*
  - *InventoryJurisdiction: Ref. to IfcActor*
  - *InventoryResponsible: Set [0:N] of Ref. to IfcActor*
  - *LastUpdateDate: IfcDate*
- ***IfcSpaceInventory subtype of IfcInventory***
  - *HasSpaces: Set [0:N] of Ref. to IfcSpaceElement*
  - *TotalSpaces: IfcInteger*
  - *TotalNetArea: IfcArea*
- ***WorkOrder of IfcTypeDefinition of DocumentType in IfcDocument***
  target object = "IfcDocument"
  shared = Att_DocumentType
  occurance = Att_WorkOrder
- ***Att_WorkOrder***
  - *TransactionCode: IfcString*
  - *RequestID: IfcString*
  - *Facility: Set [1:N] of Ref. to IfcProductObject*
  - *DateOfRequest: IfcDate*
  - *ShortJobDescription: IfcString*
  - *JobDescription: Set [0:N] of IfcString*
  - *Justification: Set [0:N] of IfcString*
  - *IfNotAccomplished: Set [0:N] of IfcString*
  - *WorkRequest: Ref. to IfcWorkTask*
  - *EstimatedCost: IfcCostSchedule*
  - *ContractualType: Enum (InHouse, SelfHelp, Contract)*
  - *Budget: Ref. to IfcBudget*
  - *RequestBy: Ref. to IfcActor*
  - *RequestTo: Ref. to IfcActor*
  - *AdditionalContact: Ref. to IfcActor*
  - *Approval: Ref. to IfcApproval*
- ***PurchaseOrder of IfcTypeDefinition of DocumentType in IfcDocument***
  target object = "IfcDocument"
  shared = Att_DocumentType
  occurance = Att_PurchaseOrder
- ***Att_PurchaseOrder***

  - *PurchaseOrder_No: IfcString*
  - *CompanyTitle: Ref. to IfcActor*
  - *SupplierName: Ref. to IfcActor*
  - *Date: IfcDate*
  - *Remark: Set [0:N] of IfcString*
  - *DateRequired: IfcString*
  - *DateScheduled: IfcDate*
  - *DateActual: IfcDate*
  - *FOB: IfcString*
  - *ShipMethod: IfcString*
  - *TotalCost: IfcCost*
  - *TotalItems: IfcInteger*
  - *PurchaseItems: List [0:N] of Att_PurchaseOrderItem*
  - *Approval: Ref. to IfcApproval*
- ***Att_PurchaseOrderItem***
  - *ItemNumber: IfcInteger*
  - *Quantity: IfcReal*
  - *Code: IfcString*
  - *Unit: IfcUnit*
  - *UnitPrice: IfcCost*
  - *TotalCost: IfcCost*
  - *InvoiceAmount: IfcCost*
  - *TotalBalance: IfcCost*
  - *InPurchaseOrder: Ref. to IfcDocument*
- ***ChangeOrder of IfcTypeDefinition of DocumentType in IfcDocument***
  target object = "IfcDocument"
  shared = Att_DocumentType
  occurance = Att_ChangeOrder
- ***Att_ChangeOrder***
  - *ChangeOrderNo: IfcString*
  - *Description: Set [0:N] of IfcString*
  - *Date: IfcDate*
  - *IssuedBy: Ref. to IfcActor*
  - *IssuedTo: Ref. to IfcActor*
  - *Approval: Ref. to IfcApproval*
- ***IfcFurnitureInventory subtype of IfcInventory***
  - *TotolValueOriginal: IfcCost*
  - *TotalValue: IfcCost*
  - *FurnitureInventory: Set [0:N] of Ref. to IfcElement*
- ***IfcEquipmentInventory subtype of IfcInventory***
  - *TotolValueOriginal: IfcCost*
  - *TotalValue: IfcCost*
  - *EquipmentInventory: Set [0:N] of Ref. to IfcEquipment*

5.10.1.3. RoadMap Issues

Interoperability Issues (see the last section of this document for more information)

Applications from which information is needed:

- *Architecture (Spaces, Walls)*
- *Construction (As-built information on walls, building systems, etc.)*

- *Electrical (Wiring, cabling)*
- *Communications (Telco, networks)*
- *HVAC System (cooling capacity, airflow, humidity, etc.)*

Applications for which information is produced:

- *Architecture (as-builts)*
- *Electrical (Wiring, cabling)*
- *Communications (Telco, networks)*
- *HVAC System (cooling capacity, airflow, humidity, etc.)*

Value of software supporting this process

- *FM: Very High (in the top 3)*
- *Architecture: High (in the top 5)*
- *CM/Cost: Very High (in the top 3)*
- *Building Service: High (in the top 5)*
- *HVAC: High (in the top 5)*

Sponsor Software Companies

The following organizations that develop software used by AEC end users have shown interest in developing applications which implement the process:

- *Naoki Systems Inc.*
- *Visio Corporation*

5.10.1.4. Issues identified in reviews

No reviews recorded

Issues

Proposed resolution

- No issues recorded
    - No resolutions recorded

5.10.2. Design of Workstations

The facility manager (also interior designers, architects, furniture manufactures and designers, contract furniture dealers, etc.) designs typical workstations to be used by office staff. The process starts from defining the functional requirements of the workstation based on the work types of the employees who use the workstation. The workstation to be designed must also meet the requirements of basic human dimensions for spaces. Special requirements such as a wheelchair must be considered. The design drawings and specifications should be produced based on the configurations of the workstation components and equipment. Final design must be approved before implementation.

5.10.2.1. Information Analysis by Task

5.10.2.1.1. Task A - Define Workstation Requirements

Define the basic components, component types and equipment, and the security, privacy and special requirements according to the employee type, work types, and company policies, etc..

Input Information:

- *Functional information*
- *Employee Information*
    - *Employee type*

- *secretarial, managerial, technical, programmer, reception, telephone respondent, etc.*
- *Number of employees who share the workstation*
- *Task Information*
  - *Work type*
    - *word processing, programming, drafting, etc.*
  - *Work load*
    - *e.g. average of weekly working hours*
    - *amount of paper files to be stored*
    - *average amount of files produced daily*
- *User requirements*
- *special requirements for the particular user, e.g. a wheelchair, special size requirement of the chairs.*
- *Company policies*
- *Max. and min. for managers or employees*

Output Information:

- *Component List*
- *Workstation component types (all following items include dimensions)*
  - *Work surfaces (including size)*
    - *writing & reading surface*
    - *computer surface*
    - *meeting surface*
    - *conference surface*
    - *reference material surface*
    - *printer surface*
  - *Storage and storage types*
    - *overhead storage*
    - *shelf storage*
    - *stationary storage*
    - *office supplies*
    - *personal items*
    - *drawers*
    - *file storage (file cabinets, file trays)*
  - *Light fixtures*
    - *ceiling-mounted lighting*
    - *task lighting*
  - *Panels*
    - *workstation partitions (glazed partitions, partitions with door, curved partitions)*
    - *screens (solid screen, glass screen)*
  - *Seating*
    - *guest seating (number of seats)*
    - *desk seating (number of seats)*
- *List of equipment types*
  - *Telephone, answering machine*
  - *electric typewriter*
  - *Computer, external modem, external CD-ROM, tape drive, speakers, etc.*
  - *Printer, fax machine, copier, scanner*
- *Workstation requirements*
- *Security requirements*
  - *drawers must lock*
  - *files must lock*
  - *fireproof file cabinet*
  - *password required to access computer*
- *Privacy requirements*
  - *speech privacy*
  - *visual privacy*
  - *equipment sharing (sharable or not)*

- *Special requirements*
  - *face-to-face interaction*
  - *files delivery requirements*
  - *aesthetic requirements*

Project Model Usage Requirements:

Existing Classes:

- ***IfcActor (used by IfcWorkstation)***
- ***IfcArea (used in IfcWorkstationCompanyPolicy)***
- ***IfcAssembledElement (superclass of IfcWorkstation)***
- ***IfcBoolean (used by Att_Storage) <not sure whether IfcBoolean exists though)***
- ***IfcBoundingBox (used by IfcWorkstation)***
- ***IfcControlObject (superclass of IfcWorkstationWorkload, IfcWorkstationRequirement, IfcWorkstationCompanyPolicy)***
- ***IfcCost (used by IfcWorkstationCompanyPolicy)***
- ***IfcFixture (used by Att_TaskLighting)***
- ***IfcInteger (used by IfcWorkstationWorkload, Att_Storage)***
- ***IfcLength (used by Att_SystemFurnitureType, Att_Worksurface, Att_Panel, Att_Storage)***
- ***IfcManufacturedElement (supertype of IfcSystemFurniture, used in IfcWorkstation)***
- ***IfcOfficeEquipment (used by IfcWorkstation)***
- ***IfcPolyCurve2D (used by IfcWorkstation, Att_Panel)***
- ***IfcReal (used by IfcWorkstationWorkload)***
- ***IfcSpace (referenced in IfcSystemFurniture)***
- ***IfcString (used by IfcWorkstationRequirement, IfcWorkstationCompanyPolicy, IfcWorkstation, Att_SystemFurnitureType, Att_Panel, Att_Worksurface, Att_Storage)***
- ***IfcTimeDuration (used by IfcWorkstationWorkload)***
- ***IfcTypeDefinition (used by IfcSystemFurniture, Panel, Worksurface, Storage)***
- ***Att_FurnitureType***
  - *<attributes defined in IFC 1.0 spec + following>*
  - *Product_code: IfcString*
  - *Width: IfcLength <nominal overall width>*
  - *Height: IfcLength <nominal overall height>*
  - *Depth: IfcLength <nominal overall depth>*
  - *Material: IfcString*
  - *Finishing: IfcString <e.g. walnut, fabric>*

New classes:

- ***IfcWorkstationWorkload   subtype of IfcControlObject***
  - *AverageWorkhourWeekly: IfcTimeDuration*
  - *TotalPaperfilesToStore: IfcInteger*
    - *<used to determine file storage>*
  - *AveragePaperfilesProducedDaily: IfcInteger*
  - *TotalComputerfiles: IfcReal*
    - *<used to determine computer equipment>*
    - *<in unit of MB>*
- ***IfcWorkstationCompanyPolicy   subclass of IfcControlObject***
  - *EmployeeType: IfcString*

    - *<e.g. manager, programmer, secretary, etc.>*
  - *MaxWorkstationSize: IfcArea*
  - *MinWorkstationSize: IfcArea*
  - *FurnitureStyle: IfcString*
  - *CostLimit: IfcCost*

- ***IfcWorkstation subclass of IfcAssembledElement***
  - *Components: Set [0:N] of Ref. to IfcManufacturedElement*
  - *<list of worksurfaces and storage, tables, chairs, etc., excluding the vertical panels>*
  - *Equipment: Set [0:N] of Ref. to IfcOfficeEquipment*
  - *<list of office equipment>*
  - *Panels: Set [0:N] of Ref. to IfcSystemFurniture*
  - *<list of furniture types, i.e. panel types>*
  - *Profile: IfcPolyCurve2D*
  - *Group: Ref. to IfcWorkstationGroup (to be defined in the next process)*
  - *<represents the workstation group that the workstation belongs to >*
  - *Has_boundingbox: IfcBoundingBox*
  - *Workload: IfcWorkstationWorkload*
  - *Company_policy: Ref. to IfcWorkstationCompanyPolicy*
  - *Assigned_to: Set [0:N] of Ref. to IfcActor*

- ***IfcSystemFurniture subtype of IfcManufacturedElement***
  - *Furniture_type: Ref. to IfcTypeDefinition*
  - *<Panel, Worksurface, Storage>*
  - *Workstation: Set [0:N] of Ref. to IfcWorkstation*
  - *Stored_in: Ref. to IfcSpace*

- ***Att_SystemFurnitureType shared AttributeSet of Furniture_type in IfcSystemFurniture***
  - *Group_code: IfcString <e.g. panels, worksurfaces, storage, etc.>*
  - *Width: IfcLength <i.e. nominal width>*
  - *Height: IfcLength <i.e. nominal length>*
  - *Finishing: IfcString <e.g. walnut, fabric>*

- ***Panel of IfcTypeDefinition of Type in IfcSystemFurniture***
  target object: IfcSystemFurniture
  shared = Att_SystemFurnitureType
  occurance = Att_Panel

- ***Worksurface of IfcTypeDefinition of FurnitureType in IfcSystemFurniture***
  target object: IfcSystemFurniture
  shared = Att_SystemFurnitureType
  occurance = Att_Worksurface

- ***Storage of IfcTypeDefinition of FurnitureType in IfcSystemFurniture***
  target object: IfcSystemFurniture
  shared = Att_SystemFurnitureType
  occurance = Att_Storage

- ***Att_Panel occurance AttributeSet for FurnitureType Panel in IfcSystemFurniture***
  - *Shape: IfcPolyCurve2D*
  - *Opening: IfcPolyCurve2D*
  - *Panel_type: IfcString*
  - *<e.g. Acoustical, Horz_Seg, Monolithic, Glazed, Open, Ends, Door, Screen, etc.>*
  - *Thickness: IfcLength*

- ***Att_Worksurface occurance AttributeSet for FurnitureType Worksurface in IfcSystemFurniture***
  - *Use_Purpose: IfcString*
  - *<e.g. writing/reading, computer, meeting, printer, reference files, etc.>*

  - *Support_type: IfcString*
  - *<i.e. Freestanding or supported>*
  - *HangingHeight: IfcLength*
  - *Thickness: IfcLength*
  - *ShapeDescription: IfcString*
  - *<corner square, rectangle, etc.>*
- ***Att_Storage   occurance AttributeSet for FurnitureType Storage in IfcSystemFurniture***
  - *IsOverhead: IfcBoolean (not sure whether IfcBoolean exists in the current version)*
  - *Support_type: IfcString*
  - *<i.e. Freestanding or supported>*
  - *Use_Purpose: IfcString*
  - *<e.g. shelf, stationary, office supplies, personal items, etc.>*
  - *Number_of_drawers: IfcInteger*
  - *HangingHeight: IfcLength <if IsOverhead>*
  - *Depth: IfcLength*
- ***Att_TaskLighting   occurance AttributeSet in IfcFixture***
  - *<<to be defined>>*
- ***IfcWorkstationRequirement   subclass of IfcControlObject***
  - *Security_Requirements: Set [0:N] of IfcString*
  - *Privacy_Requirements: Set [0:N] of IfcString*
  - *Special_Requirements: Set [0:N] of IfcString*

5.10.2.1.2. Task B - Determine Basic Workstation Spaces

Define spaces of the workstation (including circulation space inside of the workstation) according to the basic requirement of human dimension standards, and company policies.

Input Information:

- *Human dimension standards (width and height)*
- *worktask zone*
- *sitting zone and chair clearance zone*
- *turnaround zone*
- *Company policies*

Output Information:

- *Space definitions*
- *circulation space*
- *workstation space dimension*

Project Model Usage Requirements:

Existing Classes:

- ***IfcString (used in IfcWorkstationZone2D)***
- ***IfcLength (used in IfcWorkstationZone2D)***

New Classes:

- ***IfcWorkstationZone2D   used by IfcWorkstation***
  - *<this class has no superclass>*
  - *Workstation_zonetype: IfcString*
  - *<e.g. worktask, circulation, chair_clearance, etc.>*
  - *Length: IfcLength*
  - *Width: IfcLength*
  - *In_workstation: Ref. to IfcWorkstation*
- ***IfcWorkstation***
  - *<attributes defined in previous steps + the following>*

- *Zones: Set [0:N] of IfcWorkstationZone2D*

5.10.2.1.3. Task C - Define Workstation Configurations

Finalize all workstation components with all detailed dimensions and material information, and spaces.

Input Information:

- *Workstation information*
- *Circulation space zone*
- *Workstation requirements*

Output Information:

- *Workstation configurations*
- *list of workstation components*
  - *types*
  - *dimensions*
  - *materials*
- *list of equipment*
  - *types*
  - *brands*
  - *dimensions*

Project Model Usage Requirements:

Existing Classes:

- ***none from this step***

New Classes:

- ***none from this step***

5.10.2.1.4. Task D - Design Workstation

Produce the workstation drawings and define the specifications according to the configurations.

Input Information:

- *Workstation configurations*

Output Information:

- *Workstation layout drawings*
  - *drawing id, drawing title, author, proof, company, etc.*
- *Workstation design specifications*
  - *Materials*
  - *Installation requirements*
  - *category id, category name, item id, item name, item description, etc.*

Project Model Usage Requirements:

Existing Classes:

- ***IfcControlObject (superclass of IfcDocument)***
- ***IfcDate (used in Att_ElectronicDocument, Att_DocumentType)***
- ***IfcInteger (used in Att_PaperDocument, Att_DocumentType, Att_Specification)***
- ***IfcProductObject (superclass of IfcDocument, used by Att_PaperDocument)***
- ***IfcReal (used in Att_ElectronicDocument, Att_Drawing)***
- ***IfcString (used in Att_ElectronicDocument, Att_DocumentType, Att_Drawing, Att_Specification)***
- ***IfcTime (used in Att_ElectronicDocument)***

- ***IfcTypeDefinition (used in IfcDocument)***
- ***IfcUnit (used in Att_Drawing)***

New Classes:

- ***IfcDocument subclass of IfcControlObject, subclass of IfcProductObject***
  - *GenericDocumentType: IfcTypeDefinition*
  - *<used to differentiate between an electronic and a paper document>*
  - *DocumentType: IfcTypeDefinition*
  - *<used to differentiate between a drawing and specification, and etc.>*
- ***ElectronicDocument of IfcTypeDefinition of GenericDocumentType in IfcDocument***
  - *target object = “IfcDocument”*
  - *shared = <none>*
  - *occurance = Att_ElectronicDocument*
- ***PaperDocument of IfcTypeDefinition of GenericDocumentType in IfcDocument***
  - *target object = “IfcDocument”*
  - *shared = <none>*
  - *occurance = Att_PaperDocument*
- ***Att_ElectronicDocument occurance AttributeSet for GenericDocumentType ElectronicDocument in IfcDocument***
  - *File_name: IfcString*
  - *FileExtension_name: IfcString*
  - *Software: IfcString*
  - *File_size: IfcReal (in unit of KB)*
  - *Directory: IftString*
  - *Backup_file: Ref. to IfcDocument*
  - *Paper_copy: Ref. to IfcDocument*
  - *Last_save_time: IfcTime*
  - *Last_save_date: IfcDate*
  - *Type: IfcString <hidden, read-only, etc.)*
- ***Att_PaperDocument occurance AttributeSet for GenericDocumentType Paper Document in IfcDocument***
  - *Location: Ref. to IfcProductObject <more appropriate if there is something like IfcRoot>*
  - *Total_pages: IfcInteger*
  - *Electronic_copy: Ref. to IfcDocument*
- ***Drawing of IfcTypeDefinition of DocumentType in IfcDocument***
  - *target object = “IfcDocument”*
  - *shared = Att_DocumentType*
  - *occurance = Att_Drawing*
- ***Specification of IfcTypeDefinition of DocumentType in IfcDocument***
  - *target object = “IfcDocument”*
  - *shared = Att_DocumentType*
  - *occurance = Att_Specification*
- ***Att_DocumentType shared AttributeSet for DocumentType in IfcDocument***
  - *Author: IfcString*
  - *Company: IfcString*
  - *Title: IfcString*
  - *Revision_Code: IfcString*
  - *Revision_Number: IfcInteger*
  - *Last_modified_date: IfcDate*
  - *First_created_date: IfcDate*
- ***Att_Drawing occurance AttributeSet for DocumentType Drawing in IfcDocument***

- *Drawing_id: IfcString*
- *Specifications: Set [0:N] of Ref. to IfcDocument*
- *Scale: IfcReal*
- *Unit: IfcUnit*
- *Related_drawings: Set [0:N] of Ref. to IfcDocument*

- ***Att_Specification occurance AttributeSet for DocumentType Specificatioin in IfcDocument***
  - *Specification_id: IfcString*
  - *General_description: IfcString*
  - *Related_drawings: Set [0:N] of Ref. to IfcDocument*
  - *Total_words: IfcInteger*

5.10.2.1.5. Task E - Approve Design

The process examines the design and attempts to approve it.

Input Information:

- *Workstation layout drawings*
- *Workstation design specifications*

Output Information:

- *Legal/Record Documents*
- *approved design*
- *<< the drawings and specs from input that are approved >>*

Project Model Usage Requirements:

Existing Classes:

- ***none from this step***

New Classes:

- ***IfcDocument***

5.10.2.2. IFC Model Impact

Usage/Extensions to R1.0 object types

- ***Att_FurnitureType***
  - *<attributes defined in IFC 1.0 spec + following>*
  - *Product_code: IfcString*
  - *Width: IfcLength <nominal overall width>*
  - *Height: IfcLength <nominal overall height>*
  - *Depth: IfcLength <nominal overall depth>*
  - *Material: IfcString*
  - *Finishing: IfcString <e.g. walnut, fabric>*
- ***IfcActor (used by IfcWorkstation)***
- ***IfcArea (used in IfcWorkstationCompanyPolicy)***
- ***IfcAssembledElement (superclass of IfcWorkstation)***
- ***IfcBoolean (used by Att_Storage) <not sure whether IfcBoolean exists though)***
- ***IfcBoundingBox (used by IfcWorkstation)***
- ***IfcControlObject (superclass of IfcDocument, IfcWorkstationWorkload, IfcWorkstationRequirement, IfcWorkstationCompanyPolicy)***
- ***IfcCost (used by IfcWorkstationCompanyPolicy)***

- ***IfcDate (used in Att_ElectronicDocument, Att_DocumentType)***
- ***IfcFixture (used by Att_TaskLighting)***
- ***IfcInteger (used by IfcWorkstationWorkload, Att_Storage)***
- ***IfcInteger (used in IfcWorkstationWorkload, Att_Storage, Att_PaperDocument, Att_DocumentType, Att_Specification)***
- ***IfcLength (used by Att_SystemFurnitureType, Att_Worksurface, Att_Panel, Att_Storage, IfcWorkstationZone2D)***
- ***IfcManufacturedElement (supertype of IfcSystemFurniture, used in IfcWorkstation)***
- ***IfcOfficeEquipment (used by IfcWorkstation)***
- ***IfcPolyCurve2D (used by IfcWorkstation, Att_Panel)***
- ***IfcProductObject (superclass of IfcDocument, used by Att_PaperDocument)***
- ***IfcReal (used in IfcWorkstationWorkload , Att_ElectronicDocument, Att_Drawing)***
- ***IfcSpace (referenced in IfcSystemFurniture)***
- ***IfcString (used by IfcWorkstationRequirement, IfcWorkstationCompanyPolicy, IfcWorkstation, Att_SystemFurnitureType, Att_Panel, Att_Worksurface, Att_Storage, Att_ElectronicDocument, Att_DocumentType, Att_Drawing, Att_Specification, IfcWorkstationZone2D)***
- ***IfcTime (used in Att_ElectronicDocument)***
- ***IfcTimeDuration (used by IfcWorkstationWorkload)***
- ***IfcTypeDefinition (used by IfcDocument, IfcSystemFurniture, Panel, Worksurface, Storage)***
- ***IfcUnit (used in Att_Drawing)***

New object types required

- ***IfcWorkstationWorkload   subtype of IfcControlObject***
  - *AverageWorkhourWeekly: IfcTimeDuration*
  - *TotalPaperfilesToStore: IfcInteger*
  - *<used to determine file storage>*
  - *AveragePaperfilesProducedDaily: IfcInteger*
  - *TotalComputerfiles: IfcReal*
  - *<used to determine computer equipment>*
  - *<in unit of MB>*
- ***IfcWorkstationCompanyPolicy   subclass of IfcControlObject***
  - *EmployeeType: IfcString*
  - *<e.g. manager, programmer, secretary, etc.>*
  - *MaxWorkstationSize: IfcArea*
  - *MinWorkstationSize: IfcArea*
  - *FurnitureStyle: IfcString*
  - *CostLimit: IfcCost*
- ***IfcWorkstation   subclass of IfcAssembledElement***
  - *Components: Set [0:N] of Ref. to IfcManufacturedElement*
  - *<list of worksurfaces and storage, tables, chairs, etc., excluding the vertical panels>*
  - *Equipment: Set [0:N] of Ref. to IfcOfficeEquipment*
  - *<list of office equipment>*
  - *Panels: Set [0:N] of Ref. to IfcSystemFurniture*
  - *<list of furniture types, i.e. panel types>*
  - *Profile: IfcPolyCurve2D*
  - *Group: Ref. to IfcWorkstationGroup (to be defined in the next process)*
  - *<represents the workstation group that the workstation belongs to >*

- *Has_boundingbox: IfcBoundingBox*
- *Workload: IfcWorkstationWorkload*
- *Company_policy: Ref. to IfcWorkstationCompanyPolicy*
- *Assigned_to: Set [0:N] of Ref. to IfcActor*
- *Zones: Set [0:N] of IfcWorkstationZone2D*

- ***IfcSystemFurniture subtype of IfcManufacturedElement***
  - *Furniture_type: Ref. to IfcTypeDefinition*
  - *<Panel, Worksurface, Storage>*
  - *Workstation: Set [0:N] of Ref. to IfcWorkstation*
  - *Stored_in: Ref. to IfcSpace*

- ***Att_SystemFurnitureType shared AttributeSet of Furniture_type in IfcSystemFurniture***
  - *Group_code: IfcString <e.g. panels, worksurfaces, storage, etc.>*
  - *Width: IfcLength <i.e. nominal width>*
  - *Height: IfcLength <i.e. nominal length>*
  - *Finishing: IfcString <e.g. walnut, fabric>*

- ***Panel of IfcTypeDefinition of Type in IfcSystemFurniture***
  target object: IfcSystemFurniture
  shared = Att_SystemFurnitureType
  occurance = Att_Panel

- ***Worksurface of IfcTypeDefinition of FurnitureType in IfcSystemFurniture***
  target object: IfcSystemFurniture
  shared = Att_SystemFurnitureType
  occurance = Att_Worksurface

- ***Storage of IfcTypeDefinition of FurnitureType in IfcSystemFurniture***
  target object: IfcSystemFurniture
  shared = Att_SystemFurnitureType
  occurance = Att_Storage

- ***Att_Panel occurance AttributeSet for FurnitureType Panel in IfcSystemFurniture***
  - *Shape: IfcPolyCurve2D*
  - *Opening: IfcPolyCurve2D*
  - *Panel_type: IfcString*
  - *<e.g. Acoustical, Horz_Seg, Monolithic, Glazed, Open, Ends, Door, Screen, etc.>*
  - *Thickness: IfcLength*

- ***Att_Worksurface occurance AttributeSet for FurnitureType Worksurface in IfcSystemFurniture***
  - *Use_Purpose: IfcString*
  - *<e.g. writing/reading, computer, meeting, printer, reference files, etc.>*
  - *Support_type: IfcString*
  - *<i.e. Freestanding or supported>*
  - *HangingHeight: IfcLength*
  - *Thickness: IfcLength*
  - *Shape_description: IfcString*
  - *<corner square, rectangle, etc.>*

- ***Att_Storage occurance AttributeSet for FurnitureType Storage in IfcSystemFurniture***
  - *IsOverhead: IfcBoolean (not sure whether IfcBoolean exists in the current version)*
  - *Support_type: IfcString*
  - *<i.e. Freestanding or supported>*
  - *Use_Purpose: IfcString*
  - *<e.g. shelf, stationary, office supplies, personal items, etc.>*
  - *Number_of_drawers: IfcInteger*
  - *HangingHeight: IfcLength <if IsOverhead>*

  - *Depth: IfcLength*
- ***Att_TaskLighting occurance AttributeSet in IfcFixture***
  - *<<to be defined>>*
- ***IfcWorkstationRequirement subclass of IfcControlObject***
  - *Security_Requirements: Set [0:N] of IfcString*
  - *Privacy_Requirements: Set [0:N] of IfcString*
  - *Special_Requirements: Set [0:N] of IfcString*
- ***IfcWorkstationZone2D used by IfcWorkstation***
  - *<this class has no superclass>*
  - *Workstation_zonetype: IfcString*
  - *<e.g. worktask, circulation, chair_clearance, etc.>*
  - *Length: IfcLength*
  - *Width: IfcLength*
  - *In_workstation: Ref. to IfcWorkstation*
- ***IfcDocument subclass of IfcControlObject***
  subclass of IfcProductObject
  - *GenericDocumentType: IfcTypeDefinition*
  - *<used to differentiate between an electronic and a paper document>*
  - *DocumentType: IfcTypeDefinition*
  - *<used to differentiate between a drawing and specification, and etc.>*
- ***ElectronicDocument of IfcTypeDefinition of GenericDocumentType in IfcDocument***
  target object = "IfcDocument"
  shared = <none>
  occurance = Att_ElectronicDocument
- ***PaperDocument of IfcTypeDefinition of GenericDocumentType in IfcDocument***
  target object = "IfcDocument"
  shared = <none>
  occurance = Att_PaperDocument
- ***Att_ElectronicDocument occurance AttributeSet for GenericDocumentType ElectronicDocument in IfcDocument***
  - *File_name: IfcString*
  - *FileExtension_name: IfcString*
  - *Software: IfcString*
  - *File_size: IfcReal (in unit of KB)*
  - *Directory: IftString*
  - *Backup_file: Ref. to IfcDocument*
  - *Paper_copy: Ref. to IfcDocument*
  - *Last_save_time: IfcTime*
  - *Last_save_date: IfcDate*
  - *Type: IfcString <hidden, read-only, etc.)*
- ***Att_PaperDocument occurance AttributeSet for GenericDocumentType Paper Document in IfcDocument***
  - *Location: Ref. to IfcProductObject <more appropriate if there is something like IfcRoot>*
  - *Total_pages: IfcInteger*
  - *Electronic_copy: Ref. to IfcDocument*
- ***Drawing of IfcTypeDefinition of DocumentType in IfcDocument***
  target object = "IfcDocument"
  shared = Att_DocumentType
  occurance = Att_Drawing
- ***Specification of IfcTypeDefinition of DocumentType in IfcDocument***
  target object = "IfcDocument"

shared = Att_DocumentType
occurance = Att_Specification

- ***Att_DocumentType shared AttributeSet for DocumentType in IfcDocument***
  - *Author: IfcString*
  - *Company: IfcString*
  - *Title: IfcString*
  - *Revision_Code: IfcString*
  - *Revision_Number: IfcInteger*
  - *Last_modified_date: IfcDate*
  - *First_created_date: IfcDate*
- ***Att_Drawing occurance AttributeSet for DocumentType Drawing in IfcDocument***
  - *Drawing_id: IfcString*
  - *Specifications: Set [0:N] of Ref. to IfcDocument*
  - *Scale: IfcReal*
  - *Unit: IfcUnit*
  - *Related_drawings: Set [0:N] of Ref. to IfcDocument*
- ***Att_Specification occurance AttributeSet for DocumentType Specificatioin in IfcDocument***
  - *Specification_id: IfcString*
  - *General_description: IfcString*
  - *Related_drawings: Set [0:N] of Ref. to IfcDocument*
  - *Total_words: IfcInteger*

- ***IfcWorkstationGroup as defined in the next process "Floor Layout of Workstations"***

5.10.2.3. RoadMap Issues

Interoperability Issues (see the last section of this document for more information)

Applications from which information is needed:

- *Architecture*
- *CM*
- *HVAC*
- *Building Service*

Applications for which information is produced:

- *Architecture*
- *CM*
- *Building Service*

Value of software supporting this process

- *Facilities Managers/Building Owners: Very High (in the top 3)*
- *Architecture: High (in the top 5)*
- *CM/Cost Est.: High (in the top 5)*
- *have indicated that the FM space and systems furniture information would be useful for building remodeling.*
- *Building Service: High (in the top 5)*

Sponsor Software Companies

The following organizations that develop software used by AEC end users have shown interest in developing applications which implement the process:

- *Naoki Systems Inc.*
- *Visio Corporation*

5.10.2.4. Issues identified in reviews

No reviews recorded

Issues

Proposed resolution

- No issues recorded
  - No resolutions recorded

5.10.3. Floor Layout of Workstations for an Open Office

The facility manager (also interior designers, architects, or furniture dealers, etc.) designs the layout of the workstations for an open office. This process is part of the entire floor furniture and equipment planning for the department(s), and occurs after typical individual workstations have been designed. CAD-based computer programs can automate the layout design process and result in cohesive, productive and suitable department offices.

The process starts from defining the employee working relationships so that closely related workstations can be adjacently assembled into workstation groups. Common departmental areas such as circulation or service areas must be considered. The adjacency relationships between the departments or workstation groups must be determined. It is usually necessary and efficient to use block plan mechanism to mark the floor area into different and big plane blocks with each representing a departmental unit, such as a research department. Workstations and groups will then be fit into certain blocks. Actual design drawings and specifications of the workstation layout will be produced based on the workstation layout configurations. The design must be approved before implementation.

5.10.3.1. Information Analysis by Task

5.10.3.1.1. Task A - Define Employee Working Relationships

Define the individual employees working interaction patterns and meeting frequencies according to the work they perform.

Input Information:

- *Employee Information*
  - *employee name and title*
  - *work types*
  - *employee roles - roles contain requirements for interactivity*
- *Employee work interaction*
  - *within department and outside department*
    - *with whom (or location)*
    - *daily frequency*
    - *average duration each meeting*

Output Information:

- *Employee working relationships (in the form of table)*
  - *department title*
  - *interaction pattern summary*

Project Model Usage Requirements:

Existing Classes:

- ***IfcActor (referenced in Att_ActorInteraction)***
- ***IfcControlObject (superclass of IfcInteraction)***
- ***IfcInteger (used in IfcInteraction)***
- ***IfcString (used in IfcInteraction, Att_ActorInteraction)***
- ***IfcTimeDuration (used in IfcInteraction)***
- ***IfcTypeDefinition (used in IfcInteraction)***

New Classes:

- ***IfcInteraction ¬ subclass of IfcControlObject***
  - *Description: IfcString*
  - *Interaction_type: IfcTypeDefinition*
  - *Frequency_daily: IfcInteger*
  - *Average_duration: IfcTimeDuration*
- ***ActorInteraction ¬ of IfcTypeDefinition of Interaction_type in IfcInteraction***
  target object = "IfcInteraction"
  shared = <none>
  occurance = Att_ActorInteraction
- ***Att_ActorInteraction ¬ occurance AttributeSet for Interaction_type ActorInteraction in IfcInteraction***
  - *Actors: Set [2:N] of Ref. to IfcActor*
  - *Locations: IfcString*

5.10.3.1.2. Task B - Define Physically Adjacent Workstation Groups

Define the functional workstation groups according to the individual employees working relationships. A group consists of one or a few different types of typical workstations that have close working relationships, frequent interactions, and perform the same kind of function.

Input Information:

- *Employee working relationships*

Output Information:

- *Workstation groups*
  - *Workstation group function name (e.g. programmer 1, marketing 2, etc.)*
  - *List of workstations*
  - *workstation group relationships (e.g. probably bubble diagram, or interaction pattern table)*

Project Model Usage Requirements:

Existing Classes:

- ***IfcAssembledElement (superclass of IfcWorkstationGroup)***
- ***IfcString (used in IfcWorkstationGroup)***

New Classes:

- ***IfcWorkstationGroup ¬ subclass of IfcAssembledElement***
  - *Functional_name: IfcString*
  - *Workstations: Set [0:N] of Ref. to IfcWorkstation*

5.10.3.1.3. Task C - Define Departmental Common Areas

Define the areas that are shared by all employees in the department, such as common circulation and conference rooms, etc.

Input Information:

- *Minimum standards from company policy or architectural group:*
- *Employee Information*
    - *types, total numbers*
    - *work types*

Output Information:

- *Common area space requirements*
    - *circulation space*
    - *coffee room space*

Project Model Usage Requirements:

Existing Classes:

- ***IfcActor (referenced by IfcWorkstationGroup)***

New Classes:

- ***IfcWorkstationGroup***
    - *<all attributes described in the previous steps + following>*
    - *Department: Ref. to IfcActor*

5.10.3.1.4. Task D - Produce, Evaluate and Optimize Candidate Block Plans

Segment large spaces for workstation groups according to the relationships between the workstation groups, and the relationships between departments in case of multiple departments. Floor geometry constraints such as column grids, ceiling grids, window grids, the space footage must be taken into consideration. A floor block can contain one or more workstation groups, or one or more workstations.

Input Information:

- *Departmental information*
    - *name, function*
- *Bubble diagram*
    - *e.g. for interdepartmental interactions*
- *Department interaction patterns*
    - *all department names, sizes*
    - *interaction leveling (primary, secondary, tertiary, insignificant, and none)*
    - *interaction directions*
- *Common area space requirements*
- *Workstation groups*
- *Building shell information*
    - *walls*
    - *column grids*
    - *ceiling grids*
    - *window grids*
    - *floor openings (e.g. access for raised floor)*
    - *space openings (e.g. exit area)*
    - *building core (e.g. elevators, restroom, etc.)*
    - *space footage*
- *Future reserved space*
    - *footage*
    - *preferred locations*
    - *least area requirements*

Output Information:

- *Floor block plan*
- *floor blocks*
    - *block id, block name, block footage*
    - *owning department*
    - *list of contained workstation groups*
    - *list of contained workstations*

Project Model Usage Requirements:

Existing Classes:

- ***IfcArea (used in IfcFloorBlock)***
- ***IfcControlObject (superclass of IfcFloorBlock)***
- ***IfcPolyCurve2D (used in IfcFloorBlock)***
- ***IfcString (used by IfcFloorBlock, Att_SpaceType)***
- ***IfcSpace (referenced in IfcFloorBlock)***
- ***SpaceType (extended by Att_SpaceType)***

New Classes:

- ***IfcFloorBlock ¬ subclass of IfcControlObject***
  - *Function_name: IfcString*
  - *Workstation_groups: Set [0:N] of Ref. to IfcWorkstationGroup*
  - *In_space: Ref. to IfcSpace*
  - *Profile: IfcPolyCurve2D*
  - *Area: IfcArea*
- ***Att_SpaceType shared AttributeSet for TypeDefinition SpaceType defined in R1.0***
  - *Space_name: IfcString*
  - *General_description: IfcString*
  - *Space_catalog: IfcString*

5.10.3.1.5. Task E - Define Floor Layout Configurations

Define all the detailed footage of all the workstations, workstation groups and departmental boundaries.

Input Information:

- *Floor block plan*
- *Common area space requirements*
- *Workstation groups*
- *Building shell information*
- *Existing inventory*
  - *furniture in store rooms*

Output Information:

- *Floor layout configurations*
  - *workstations*
    - *id, name, owning department, and user*
    - *footage*
  - *workstation groups*
  - *departmental boundaries*

Project Model Usage Requirements:

Existing Classes:

- ***IfcArea (referenced in IfcWorkstationGroup)***
- ***IfcManufacturedElement (referenced by IfcWorkstationGroup, etc.)***
- ***IfcOfficeEquipment (referenced by IfcWorkstationGroup)***
- ***IfcPolyCurve2D (referenced in IfcWorkstationGroup)***
- ***IfcSpace (referenced in IfcWorkstationGroup)***

New Classes:

- ***IfcWorkstationGroup extended from first step***
  - *{{all items described in previous steps + the following}}*

- *In_floor_block: Ref. to IfcFloorBlock*
- *In_space: Ref. to IfcSpace*
- *Profile: IfcPolyCurve2D*
- *Total_area: IfcArea*
- *Shared_furniture: Set [0:N] of Ref. to IfcManufacturedElement*
- *<shared furniture is not part of any workstations in the workstation group, e.g. a table for supporting a shared printer>*
- *Shared_equipment: Set [0:N] Ref. to IfcOfficeEquipment*
- *<shared equipment is not part of any workstations in the workstation group, e.g. a shared printer>*
- *Workstation_groups: Set [0:N] of Ref. to IfcWorkstationGroup*
- *<a workstation group can contain other groups>*

5.10.3.1.6. Task F - Design Floor Layout

Produce the workstation layout drawings and define the specifications.

Input Information:
- *Floor layout configurations*
- *Existing inventory*
- *Architectural layout*

Output Information:
- *Floor layout drawings*
  - *drawing id, name, author, proof, company, etc.*
- *Floor layout specifications*
  - *Materials*
  - *Installation requirements*

Project Model Usage Requirements:

Existing Classes:
- ***<none in this step>***

New Classes:
- ***IfcDocument as defined in the last step of the process "Design of Workstations"***

5.10.3.1.7. Task G - Implement

After the design has been approved, implement the design based on the drawings and specifications.

Input Information:
- *Existing inventory*
- *Floor layout drawings*
- *Floor layout specifications*

Output Information:
- *As-built drawings*

Project Model Usage Requirements:

Existing Classes:
- ***<none in this step>***

New Classes:
- ***IfcSpaceInventory (as defined in the 1st process: Occupancy Planning)***
- ***IfcFurnitureInventory (as defined in the 1st process: Occupancy Planning)***
- ***IfcEquipmentInventory (as defined in the 1st process: Occupancy Planning)***

5.10.3.1.8. Task H - Update

This is an on-going process that occurs during the course of design implementation. Inventories are updated. Drawings are changed and as-built drawings are produced overtime.

Input Information:
- *Existing inventory*
- *As-built drawings*

Output Information:
- *Updated inventories*

Project Model Usage Requirements:

Existing Classes:
- ***<none in this step>***

New Classes:
- ***<none in this step>***

5.10.3.2. IFC Model Impact

Usage/Extensions to R1.0 object types

- ***IfcActor (referenced by IfcWorkstationGroup, Att_ActorInteraction)***
- ***IfcArea (referenced in IfcWorkstationGroup, IfcFloorBlock)***
- ***IfcAssembledElement (superclass of IfcWorkstationGroup)***
- ***IfcControlObject (superclass of IfcFloorBlock, IfcInteraction)***
- ***IfcInteger (used in IfcInteraction)***
- ***IfcManufacturedElement (referenced by IfcWorkstationGroup, etc.)***
- ***IfcOfficeEquipment (referenced by IfcWorkstationGroup)***
- ***IfcPolyCurve2D (referenced in IfcWorkstationGroup, IfcFloorBlock)***
- ***IfcSpace (referenced in IfcFloorBlock, IfcWorkstationGroup)***
- ***SpaceType (extended by Att_SpaceType)***
- ***IfcString (used by IfcFloorBlock, Att_SpaceType, IfcInteraction, Att_ActorInteraction, IfcWorkstationGroup)***
- ***IfcTimeDuration (used in IfcInteraction)***
- ***IfcTypeDefinition (used in IfcInteraction)***

New object types required

- ***IfcInteraction ¬ subclass of IfcControlObject***
  - *Description: IfcString*
  - *Interaction_type: IfcTypeDefinition*
  - *Frequency_daily: IfcInteger*
  - *Average_duration: IfcTimeDuration*
- ***ActorInteraction ¬ of IfcTypeDefinition of Interaction_type in IfcInteraction***
  target object = "IfcInteraction"
  shared = <none>
  occurance = Att_ActorInteraction
- ***Att_ActorInteraction ¬ occurance AttributeSet for Interaction_type ActorInteraction in IfcInteraction***

- *Actors: Set [2:N] of Ref. to IfcActor*
- *Locations: IfcString*

- ***IfcWorkstationGroup ¬ subclass of IfcAssembledElement***
  - *Functional_name: IfcString*
  - *Workstations: Set [0:N] of Ref. to IfcWorkstation*
  - *Department: Ref. to IfcActor*
  - *In_floor_block: Ref. to IfcFloorBlock*
  - *In_space: Ref. to IfcSpace*
  - *Profile: IfcPolyCurve2D*
  - *Total_area: IfcArea*
  - *Shared_furniture: Set [0:N] of Ref. to IfcManufacturedElement*
  - *<shared furniture is not part of any workstations in the workstation group, e.g. a table for supporting a shared printer>*
  - *Shared_equipment: Set [0:N] Ref. to IfcOfficeEquipment*
  - *<shared equipment is not part of any workstations in the workstation group, e.g. a shared printer>*
  - *Workstation_groups: Set [0:N] of Ref. to IfcWorkstationGroup*
  - *<a workstation group can contain other groups>*
- ***IfcFloorBlock ¬ subclass of IfcAssembledElement***
  - *Function_name: IfcString*
  - *Workstation_groups: Set [0:N] of Ref. to IfcWorkstationGroup*
  - *In_space: Ref. to IfcSpace*
  - *Profile: IfcPolyCurve2D*
  - *Area: IfcArea*
- ***Att_SpaceType shared AttributeSet for TypeDefinition SpaceType defined in R1.0***
  - *Space_name: IfcString*
  - *General_description: IfcString*
  - *Space_catalog: IfcString*
- ***IfcDocument as defined in the last step of the process "Design of Workstations"***
- ***IfcWorkstation as defined in the process "Design of Workstations"***
- ***IfcSpaceInventory (as defined in the 1st process: Occupancy Planning)***
- ***IfcFurnitureInventory (as defined in the 1st process: Occupancy Planning)***
- ***IfcEquipmentInventory (as defined in the 1st process: Occupancy Planning)***

5.10.3.3. RoadMap Issues

Interoperability Issues (see the last section of this document for more information)

Applications from which information is needed:

- *Architecture*
- *CM*
- *HVAC*
- *Building Service*

Applications for which information is produced:

- *Architecture*
- *CM*
- *Building Service*

Value of software supporting this process

- *Facilities Managers/Building Owners: Very High (in the top 3)*
- *Architecture: High (in the top 5)*
- *CM/Cost Est.: High (in the top 5)*
- *have indicated that the FM space and systems furniture information would be useful for building remodeling.*
- *Building Service: High (in the top 5)*
- *HVAC:*

Sponsor Software Companies

The following organizations that develop software used by AEC end users have shown interest in developing applications which implement the process:

- *Naoki Systems Inc.*
- *Visio Corporation*

5.10.3.4. Issues identified in reviews

No reviews recorded

Issues

Proposed resolution

- No issues recorded
    - No resolutions recorded

5.10.3.5. Information needed by FM from other phases

FM-4 is currently concerned with only three processes

- Occupancy planning
- Design of Workstations
- Layout of Workstations

The info requirements listed below is not limited to these three processes

5.10.3.5.1. Building Systems Group

Building systems performance

- Performance characteristics of all mechanical systems (as level of detail typically specified by architect)

HVAC - Air exchanges per hour

- Temperature ranges
- Controls/location/zones
- Operating/maintenace instructions (Also from contractor)
- Solar performance

Mechanical

- restroom/wastewater capacities (also from architect for non mechanical aspects)

Electrical

- Circuit capacities
- locations
- switching

Energy use

- Average energy use
- Control of energy use

Elevator and other vertical transport systems

- Capacities
- Operational charateriatics
- Special features/uses

Fire protection

- Zoning/fire ratings
- Location/type

Handicapped features/requirements

- All operational manuals, maintenance, warrantee, service contact, inspection info (Also from architecture group)

5.10.3.5.2. Architecture Group

A. General Building Information

- Programming info
- "Owners intent"
- When built, Cost, Sq ft.
- Usable, rentable, common space
- Historical issues
- Utilities
  - *Suppliers, contacts*
  - *main controls, shut-off valves, etc.*
- Landscaping & Grounds management
  - *Irrigation system (plan and controls)*
  - *Surface water management' (strom water detention)*

B. Equipment details (Also form mechanical group)

- Operation/training-manuals, materials, schedules
- Warrantee, contacts
- Service, customer support, parts & supplies
- Connectivity, Redundancy (e.g. is this the only one)

C. Site and Location Information

- Transportation, access, parking
  - *Registered transportation management plan*
- Disaster plans, flood plain issues
- Community services
  - *Police, traffic enforcement*
  - *EPA office*
  - *Fire station*
  - *Medical*
  - *News media*

D. Building shell information

- Wall, window wall, roofing

E. Systems

- Status and Condition information
- Maintenance records/ schedules
- Replacements- required, planned, desired
- Required service
- Signage plans / guidelines

F. Occupancy

- Operating costs, issues
- Leasing, renting
- Taxes
- Sanitary service
- Water utilities, storm water management
- Insurance

- Local gov inspections/assessments
- Security

G. Communication & Computing

- Telephone/ telecomunications contracts/services
- Service providers/ acess/ capacity
- Networking & cabling

H. Hazardous materials, waste

- Management
- Compliance

I. Space Information

- 1) Existing architectural layout and building components (“as-builts”)
  - *Walls, doors, comfort controls and zones, floors, signing,*
  - *finishes, colors*
- 2) Architectural performance
  - *Architectural rationale (spaces, lighting, colors - how they “work”)*
  - *Illumination*
  - *Sound/accoustics*
- 3) Structural (from structural group)
  - *Structural elements and layout*
  - *Layman description of how the structure works, what is important*
  - *Siesmic design stds*
  - *Design loads*
- 4) Architectural Maintenance
  - *Schedule*
  - *Cost*
- 5) Mechanical & Ele trical (related to spaces)
  - *Elements, layout, capacities*
- 6) Safety
- 7) Security
  - *Penetrations, alarms, monitors*
  - *Locks, keyplans*
- 8) Access and escape
- 9) Use and code guidelines
- 10) What is unusual, what is important?

5.11. [SI-1] Photo Accurate Visualization

5.11.1. Photo Accurate Visualization

Visualization is performed by architects, lighting engineers and renderers with computer and electronic visualization skills. It can be used at any point in the building, lighting or interior design process, as well as during the occupancy of the building. Three-dimensional representation of a space or a building is the starting point; information about surface materials and sources of light is required for the simulation. The former can be obtained from manuals, manufacturers’ catalogues, databases, etc. The later is available in technical literture and from specialised computer models. The resulting images (renderings or animations) and data (luminance) can be used for many purposes: communication about the “looks” of a design solution to making design and engineering decisions.

5.11.1.1. Information Analysis by Task

In the design of a building or other structure, the architect or designer may want to see what the building or the structure will look like, or may want to render images for the client's benefit. Such visualization may be desired at any time from the earliest architectural design or retrofitting to the final interior design.

Visualization is the key to solving lighting and daylighting design problems, and is also important in assessing building performance and human comfort issues. IFC support of this process may reduce input preparation time by 75-85% process (through automatic acquisition of building geometry and all surface properties) and thus make the use of the corresponding applications economically feasible.

5.11.1.1.1. Task A - Select Surface Materials

Select materials to associate with building surfaces, and define material properties such as reflectance, transmittance, colors, patterns, texture, etc. Input is the three-dimensional building model; output are the materials associated with each modeled surface or solid.

Input Information:
- *Three-dimensional building model*

Output Information:
- *Materials associated with each 3-d surface (list)*

1.1.1.1.1 Task B - Specify Lighting

In order to perform a valid visual simulation, it is necessary to select and place light sources (luminaires) and specify daylight conditions. Luminaires may be selected from manufacturer's catalogs, and the sun and sky conditions may be taken from a set of quantitative models appropriate to the building site.

Input Information:
- *Light source configuration and distribution data*
- *Daylight models for this site*

Output Information:
- *Positions and types of light sources (drawings and specs)*
- *Sky distribution, solar position (fromtime of day and year)*

1.1.1.1.2 Task C - Rendering

Compute 2-dimensional images for visual evaluation.

Input Information:
- *Selected views for this model*
- *Animation path (optional)*

Output Information:
- *Two-dimensional color images (floating point)*
- *Luminance and isolux contour plots (optional)*
- *Animations (optional)*

1.1.1.1.3 Task D - Evaluate Results

Once one or more images have been produced, it is often desirable to go back and iterate on the material selection and/or light source selection and placement

Project Model Usage Requirements:

Existing Classes:
- ***all that define the geometry of the space or building in the simulation***

New Classes:
- ***Light source***
- ***Surface***

5.11.1.2. IFC Model Impact

Usage/Extensions to R1.0 object types

- ***IfcMaterialLayer***
  - *bidirectional scattering distribution function (BSDF) or model thereof includes: spectral reflectance and transmittance, specularity and roughness*
  - *polarization properties*

New object types required

- ***IfcLightSource***
  - *Spectral powerdistribution (lamp)*
  - *Luminaire geometry*
  - *Photometric output distribution*
- ***IfcSurface***
  - *General shape (e.g., polygon, sphere, etc.)*
  - *Dimensions*
  - *Material and parameterization*

5.11.1.3. RoadMap Issues

Interoperability Issues

Applications from which information is needed:

- *CAD software*

Applications for which information is produced:

- *to be determined*

Value of software supporting this process

- *{{In this section, please allow for the other domains to rank your process in order of precedance for their domain, this allows us to examine the issue on a group as well as an individual level}}*
- *{{ discipline 1 }} - {{value from 1-10, 1 being the highest value, 10 being the lowest]]*
- *{{ discipline 2 }} - {{value from 1-10]]*

Sponsor Software Companies

The following organizations that develop software used by AEC end users have shown interest in developing applications which implement the process:

- *LBNL (Radiance 3.0)*
- *Lightscape Technologies*
- *Arris Integra*
- *3DStudio (rendering)*
- *Pixar (Renderman)*
- *Lightworks*
- *others*

5.11.1.4. Issues idendified in reviews

No reviews recorded

Issues

Proposed resolution

- No issues recorded
    - No resolutions recorded

5.12. [XM-2] Project Document Management

5.12.1. Project Document Management

5.12.1.1. Information Analysis by Task

Project Document Management refers to all information pertaining to the documents used to estimate, bid, purchase, and manage the building process as well as for use within the Facilities Management domain. This data identifies the document, the author of the document, changes to the document since the last change, and relationships to other documents.

- *Who performs this process? All software vendors that use drawings, specifications, and sketches during the life cycle of a project. This would include CAD, estimating, scheduling, management, and facilities management software vendors.*
- *When in the project lifecycle it is performed? From the very inception of the project, where these documents are used to define the project, through the construction of the project with all of its changes, through the management of the "building" once the project is complete.*
- *What other processes does it relate to (input from/output to/controlled by)? This process starts in the creation and modification of the documents and outputs to all processes that use the documents as a means of identification. This would include estimating where changes to the work are usually quantified by document, management, where the documents are used to control the flow of work on a project and establish what is being built by document, and Facilities Management, where documents are the prime method of identifying actual conditions in a facility.*

For Contract Drawings and Sketches, the Architect start this process during the creation of the drawings by entering information regarding the drawings. This information would include:

- Document type
- Document Id
- Description
- Document Date
- Revision Number
- Revision Date
- Document Type and Id of related documents. This might include relationships between drawings and sketches, or even objects in the drawings with objects in a sketch or maybe the object as identified within a specification.
- Document Author
- Document Revision Author
- Bulletin/Addenda reference
- Related Documents, Sections, Details, Objects

In addition, the creation of objects onto the drawing will also trigger the saving of information regarding the objects. This process is handled by the CAD software. The information saved would include:

The List of Objects

- Original
- Added
- Deleted
- Modified
- Dates for all of these
- Revisions for all of these
- Possible relationships between the objects of this drawing and the objects on other drawings, specifications, and sketches created by the Architect.

Similar information would be required for implementation of the same for Specifications. These will not be modeled here.

Once the information is provided to the document, software using drawings can take advantage of the information to organize the processes of change throughout a project as well as the interconnection of information between project contract documents, such as between drawings, drawing sections, drawing details, sketches, and specification sections.

5.12.1.1.1. Task A - Identifications Supplied By Author

The Architect or author of the document provides information regarding the document he/she is creating.

Input Information:

- *The document type, whether it is a drawing, specification, sketch, etc.*
- *The document details such as Drawing Number, Drawing Date, Author, etc.*

Project Model Usage Requirements:

To be determined

5.12.1.1.2. Task B - Identifications Supplied by Vendor

The CAD or Specifications Software Vendor (or Author?)would assign an identification and information regarding those objects within the document.

Input Information:

- *Object Types*
- *Object Specific information such as Object Identification, Creation Date, Author, etc.*

Project Model Usage Requirements:

To be determined

5.12.1.1.3. Task C - Document Modifications

The CAD or Specifications Software Vendor (or Author?) would keep track of modifications made to the documents with respect to Revision and date.

Input Information:

- *Type of Modification, such as created, modified, deleted*
- *Modification Details, such as Creation Date, Author of Change, Change Identifications*

Project Model Usage Requirements:

To be determined

5.12.1.1.4. Task D - Identify Relationships with Other Documents

A link is then made to the appropriate documents, where information is contained not within the current document.

Input Information:

- *Document Type*
- *Document Identification*

- *Internal Document Reference*

output Information:

- *Document Identification*
- *Document Type*
- *Internal Document Reference*
- *Change Information*

Project Model Usage Requirements:

To be determined

5.12.1.1.5. Task E - Estimating

Estimating software packages can now use the information provided above to estimate changes to the project on a document by document and change by change basis. This would include changes made over multiple documents since the change identification can be identical between documents.

output Information:

- *Document Type*
- *Document Identification*
- *Internal Document Reference*
- *Change Information*

Project Model Usage Requirements:

To be determined

5.12.1.1.6. Task F - Scheduling

Scheduling software packages can now use the information provided above to estimate changes to the project schedule on a document by document and change by change basis. This would include changes made over multiple documents since the change identification can be identical between documents.

output Information:

- *Document Type*
- *Document Identification*
- *Internal Document Reference*
- *Change Information*

Project Model Usage Requirements:

To be determined

5.12.1.1.7. Task G - Project Management

Project Management software packages can now use the information provided above to estimate changes to the project on a document by document and change by change basis. This would include changes made over multiple documents since the change identification can be identical between documents.

output Information:

- *Document Type*
- *Document Identification*
- *Internal Document Reference*
- *Change Information*

Project Model Usage Requirements:

To be determined

5.12.1.2. IFC Model Impact

Extensions to R1.5.1 object types

To be determined

New object types

Document Type Object
Document Object

5.12.1.3. RoadMap Issues

Interoperability issues

Applications from which information is needed:

- *Architects*
- *CAD Software*
- *Engineers (those who create contract documents)*
- *Facilities Management*
- *Specifications Software Vendors*

Applications for which information is produced:

- *Owners*
- *Architects*
- *Engineers*
- *Construction Professionals*
- *Estimators*
- *Purchasers*
- *Facilities Management*

Value of software supporting this process

- *Construction Professionals {I consider this my highest priority - a definite 10}*

Sponsor Software Companies

The following organizations that develop software used by AEC end users have shown interest in developing applications which implement the process:

- *Frontrunner, LLC*
- *Turner Corporation Internal Development*
- *Autodesk*
- *Bentley Systems (preliminary interest)*

5.12.1.4. Issues idendified in reviews

No reviews recorded

Issues

Proposed resolution

- No issues recorded
  - No resolutions recorded

6. Object Type Definition Tables

6.1. [AR-1] Architectural Model Extensions

6.1.1. Object Types

6.1.1.1. Building Shell Design

		Object Type Name	Interface name		OPT INV DER					
Subtype of				Attribute / Relation name		Data Type/Related Object Type	Min	Max	Default	Definition
IfcBuildingElement (an assembly, defined using IfcRelAssemblesElements)	1	**IfcCurtainWall**								Exterior wall of a building which is an assembly of components, hung from the edge of the floor/roof structure rather than bearing on a floor
			I_CurtainWall : ConstructionDetails, SpecSection							
				ConstructionDetails		LIST [0:?] OF IfcDocumentReference	n/a	n/a	empty list	List of references to a detail drawings
				SpecSection	OPT	IfcDocumentReference	n/a	n/a	NIL	Document reference to specification section
IfcBuildingElement	2	**IfcCurtainWallElement**								Component in an building curtain wall
			I_CurtainWallElement : GenericType							
				GenericType		IfcCurtainWallElementTypeEnum	GlazingPanel	OverhangShade	GlazingPanel	Generic type which keys to type definition for all primary types of elements in Curtain walls. Anything missed will be represented using IfcProxy.
IfcBuildingElement	3	**IfcPermeableOpeningCover**								Permeable cover for an opening which allows airflow (definition BS 6100)
			I_PermeableOpeningCover : GenericType							
				GenericType		IfcPermeableOpeningCoverTypeEnum	Grate	Screen	Screen	Predefined generic types are specified in an Enum. Type driven Psets are defined for each generic type (as the required attributes differ). The GenericType for a given instance drives determines the type of Pset attached at runtime throug the associated TypeDef object (defined at the IfcObject supertype).
IfcFloor	4	**IfcRamp**								Inclined floor surface
			I_Ramp : Length, Width, Slope, Landings, Railings, ConstructionDetails, SpecSection							
				Length		IfcPositiveLengthMeasure	0.0	see type	0.0	length of ramp
				Width		IfcPositiveLengthMeasure	0.0	see type	0.0	width of ramp
				Slope		IfcAngleMeasure	0.0	see type	0.0	slope of ramp - relative to horizontal (non-sloping) floor
				Landings		LIST [0:2] OF Ref [IfcStairRampLanding]	n/a	n/a	empty list	References to landing objects that are either end of ramp.
				Railings		LIST [0:?] OF Ref [IfcRailing]	n/a	n/a	empty list	List of reference railings (either handrails or guardrails) for this ramp
				ConstructionDetails		LIST [0:?] OF Ref [IfcDocumentReference]	n/a	n/a	empty list	List of references to drawing documents which define construction details (especially dealing with drainage)
				SpecSection		Ref [IfcDocumentReference]	n/a	n/a	NIL	Reference to a section in the construction specifications

6.1.1.2. Building Core Design

*** None defined for this process ***

6.1.1.3. Stair Design

Subtype of		Object Type Name	Interface name	Attribute / Relation name	OPT INV DER	Data Type/Related Object Type	Min	Max	Default	Definition
IfcBuildingElement (Assembly using IfcRelAssemblesElements)	9	**IfcStair**								Assembly of building components allowing occupants to walk (step) from Floor (or Landing) to another at a different elevation.
			I_Stair : GenericType, EnclosingSpace, Protection							
			I_StairComponents(Landings, Flights							
				GenericType		IfcStairTypeEnum	FireStair	Standard AccessStair	Standard AccessStair	Predefined generic types are specified in an Enum. Type driven Psets are defined for each generic type (as the required attributes differ). The GenericType for a given instance drives determines the type of Pset attached at runtime throug the associated TypeDef object (defined at the IfcObject supertype).
				EnclosingSpace	OPT	Ref [IfcSpace]	n/a	n/a	NIL	Reference to the Space object in which this stair is enclosed (if any)
				Protection		BOOLEAN	FALSE	TRUE	FALSE	selection of protected or not protected stair set
				Landings		LIST [0:?] OF Ref [IfcStairRampLanding]	n/a	n/a	empty list	list relationships - landings included in this stair assembly
				Flights		LIST [1:?] OF Ref [IfcStairFlight]	n/a	n/a	empty list	list relationships - flights included in this stair assembly
IfcBuildingElement (Assembly using IfcRelAssemblesElements)	10	**IfcStairFlight**								Assembly of building components in a single "run" of stair steps (not interrupted by a landing). Also includes stringers, handrails, guardrails, etc.
			I_StairFlight : HeadRoom, Steps, Stringers, Railings, Landings							
				HeadRoom		IfcPositiveLengthMeasure	0.0	see type	0.0	Headroom clearence
				Steps		LIST [1:?] OF IfcStairStep	n/a	n/a	one step minimum	List of references to Stair Steps objects
				Stringers		LIST [0:?] OF IfcBeam	n/a	n/a	empty list	List of references to stringers for this flight. Note: stringers are a type of Beam
				Railings		LIST [0:?] OF IfcRailing	n/a	n/a	empty list	List of references to handrails and guardrails
				Landings		LIST [0:2] OF Ref [IfcStairRampLanding]	n/a	n/a	empty list	list relationships - landings included in this stair assembly
IfcBuildingElement	11	**IfcStairStep**								Individual step (riser + tread) within a stair flight. Allows human occupants to ascend or descend from one floor to another (at different elevations).
			I_StairStep : RiserHeight, TreadDepth, TreadMaterial, NosingMaterial, ConstructionDetail							
				RiserHeight		IfcPositiveLengthMeasure	0.0	30 cm	0.0	Distance from tread to tread
				TreadDepth		IfcPositiveLengthMeasure	0.0	30 cm	0.0	Distance from the front of the tread to back of the tread
				TreadMaterial		INTEGER	1	see type	1	Composition of tread. Index into the IfcMaterialList defined at the IfcBuildingElement supertype
				NosingMaterial		INTEGER	1	see type	1	Composition of tread. Index into the IfcMaterialList defined at the IfcBuildingElement supertype
				ConstructionDetail		Ref [IfcDocumentReference]	n/a	n/a	NIL	Reference to construction detail drawing
IfcFloor	12	**IfcStairOrRampLanding**								Floor section to which one or more stair flights connects. May or may not be adjacent to a building storey floor.
			I_StairOrRampLanding : PartOfStair, ConnectedFlights, HeadRoom, ConstructionDetails, Railings							

				PartOfStair	INV	Ref [IfcStair]	n/a	n/a	NIL	reference to the stair for which this landing is a component (inverse for Landings)
				ConnectedFlights	INV	LIST [0:?] OF Ref [IfcStairFlight]	n/a	n/a	empty list	list of Stair Flights connected to this landing
				HeadRoom		IfcPositiveLengthMeasure	0.0	see type	0.0	Headroom clearence
				ConstructionDetail		LIST [0:?] OF Ref [IfcDocumentReference]	n/a	n/a	NIL	Reference to construction detail drawing
				Railings		LIST [0:?] OF IfcRailing	n/a	n/a	empty list	List of references to handrails and guardrails for this landing
Subtype from IfcBuiltIn	13	**IfcRailing**								
			I_Railing : GenericType, RailingHardware							
				GenericType		IfcRailingTypeEnum	Handrail	Balustrade	Handrail	Predefined generic types are specified in an Enum. Type driven Psets are defined for each generic type (as the required attributes differ). The GenericType for a given instance drives determines the type of Pset attached at runtime throug the associated TypeDef object (defined at the IfcObject supertype).
				RailingHardware		LIST [0:?] OF IfcBuiltInAccessory	n/a	n/a	empty list	List of references to accessory/mounting hardware for this railing.
Subtype from IfcBuiltIn	14	**IfcCabinet**								
			I_Cabinet : GenericType, CabinetHardware							
				GenericType		IfcCabinetTypeEnum	Bathroom	Office	Bathroom	Predefined generic types are specified in an Enum. Type driven Psets are defined for each generic type (as the required attributes differ). The GenericType for a given instance drives determines the type of Pset attached at runtime throug the associated TypeDef object (defined at the IfcObject supertype).
				CabinetHardware		LIST [0:?] OF IfcBuiltInAccessory	n/a	n/a	empty list	List of references to accessory hardware for this cabinet.
Subtype from IfcBuiltIn	15	**IfcCounterOrShelf**								
			I_CounterOrShelf : GenericType, CounterOrShelfHardware							
				GenericType		IfcCouterOrShelfTypeEnum	Bathroom Counter	Shelf	Shelf	Predefined generic types are specified in an Enum. Type driven Psets are defined for each generic type (as the required attributes differ). The GenericType for a given instance drives determines the type of Pset attached at runtime throug the associated TypeDef object (defined at the IfcObject supertype).
				CounterOrShelfHardware		LIST [0:?] OF IfcBuiltInAccessory	n/a	n/a	empty list	List of references to accessory hardware for this counter or shelf.

6.1.1.4. Public Toilet Design

		Object Type Name	Interface name		OPT INV DER					
Subtype of				Attribute / Relation name		Data Type/Related Object Type	Min	Max	Default	Definition
IfcBuildingElement	16	**IfcPlumbingFixture**								
			I_PlumbingFixture : GenericType							
				GenericType		IfcPlumbingFixtureType	Faucet	Dishwasher	Faucet	Predefined generic types are specified in an Enum. Type driven Psets are defined for each generic type (as the required attributes differ). The GenericType for a given instance drives determines the type of Pset attached at runtime throug the associated TypeDef object (defined at the IfcObject supertype).
IfcBuildingElement	17	**IfcElectricalFixture**								

			I_ElectricalFixture : GenericType							
				GenericType		IfcElectricalFixtureTypeEnum	Light	RadiantHeater	Light	Predefined generic types are specified in an Enum. Type driven Psets are defined for each generic type (as the required attributes differ). The GenericType for a given instance drives determines the type of Pset attached at runtime throug the associated TypeDef object (defined at the IfcObject supertype).
IfcBuiltIn	18	**IfcBuiltInAccessory**								Building hardware or attached occupant accessory - attached to one or more building elements
			I_BuiltInAccessory : GenericType, MountingHeight, MountingType, ConstructionDetails, SpecSection							
				GenericType		IfcBuiltInAccessoryTypeEnum	DoorsAndWindows	CounterOrShelfHW	Bathroom	Predefined generic types are specified in an Enum. Type driven Psets are defined for each generic type (as the required attributes differ). The GenericType for a given instance drives determines the type of Pset attached at runtime throug the associated TypeDef object (defined at the IfcObject supertype).
				MountingHeight		IfcPositiveLengthMeasure	0.0	see type	0.0	height at which the item gets connect to the wall. Value of 0.0 means this property not set.
				MountingType		STRING	n/a	n/a	empty string	Description of the method for mounting
				ConstructionDetails		LIST [0:?] OF ObjectReference (IfcDocumentReference)	n/a	n/a	empty list	List of reference to construction detail drawings
				SpecSection		ObjectReference (IfcDocumentReference)	n/a	n/a	NIL	Reference to a section of the construction specification

6.1.1.5. Roof Design

		Object Type Name	**Interface name**		OPT INV DER					
Subtype of				Attribute / Relation name		Data Type/Related Object Type	Min	Max	Default	Definition
IfcBuildingElement (assembly using IfcRelAssembles)	5	**IfcRoof**								A description of the total roof
			I_Roof : GenericType, FireRating, AtticSpace							
			I_RoofComponents : RoofSlabs, RoofFrames							
			I_RoofSurface : calc_TotalRoofSurfaceArea							
			I_RoofDrainage : PrimaryDrainage, SecondaryDrainage							
Interface			**I_Snow : SnowZone, DesignSnowLoad**							
				GenericType		IfcRoofTypeEnum	Flat	OtherSloped	Flat	Predefined generic types are specified in an Enum. Type driven Psets are defined for each generic type (as the required attributes differ). The GenericType for a given instance drives determines the type of Pset attached at runtime throug the associated TypeDef object (defined at the IfcObject supertype).
				FireRating		IfcTimeDurationMeasure	see type	see type	60 min	Time duration for fire resistence the roof assembly is rated
				AtticSpace		Ref [IfcSpace]	n/a	n/a	NIL	Reference to a Space object - the untreated attic space under this roof. Used for thermal performance calculations.
				RoofSlabs		LIST [0:?] OF Ref [IfcRoofSlab]	n/a	n/a	empty list	list of references to roof surface objects (IfcRoofslab)
				RoofFrames		LIST [0:?] OF Ref [IfcRoofFrame]	n/a	n/a	empty list	list of references to roof frame objects (IfcRoofFrame)

			calc_TotalRoofSurfaceArea		IfcAreaMeasure	see type	see type	0	Total surface area of the roof. Note: this is a calculated value, based on all of the roofslabs included in this roof. A value of 0.0 means the value has not been calculated.
			PrimaryDrainage		LIST [0:?] OF Ref [IfcDistributionElement]	n/a	n/a	empty list	List of references to all Primary drains
			SecondaryDrainage		LIST [0:?] OF Ref [IfcDistributionElement]	n/a	n/a	empty list	List of references to all secondary drains (or scuppers)
			SnowZone		STRING	na	na	empty string	Zone indicating average number of inches accumulated
			DesignSnowLoad		IfcMassMeasure	0.0	see type	0.0	Weight of design snow load. Value of 0.0 means property not set.
		IfcRoofSlab							
		I_RoofSlab : GenericType, LayerInformation, RoofingMaterial, FireRating							
			GenericType		IfcRoofSlabTypeEnum	SolidRoofSlab	ElementedRoofSlab	LayeredRoofSlab	Predefined generic types are specified in an Enum. Type driven Psets are defined for each generic type (as the required attributes differ). The GenericType for a given instance drives determines the type of Pset attached at runtime throug the associated TypeDef object (defined at the IfcObject supertype).
			LayerInformation		IfcMaterialLayerSetUsage				References a MaterialLayerSet, defines the offset relative to the extrusion path and defines the ordering of layers (left to right or right to left).
			RoofingMaterial		INTEGER	1	N	1	Index into the list of material Layers - to the roofing material
			FireRating		IfcTimeDurationMeasure	see type	see type	60 min	Time duration for fire resistence the roof assembly is rated
		IfcMaterial							
		I_Material : MaterialName, MaterialClassification, RegisteredBy							
		I_SurfaceProperties : SurfaceReflectivity, SurfaceColor, SurfaceTexture							
		I_ThermalProperties : ThermalCoefficient							
			MaterialName		STRING				
			MaterialClassification		IfcClassificationList				
			RegisteredBy	INV	IfcProjectMaterialRegistry				
			SurfaceReflectivity		IfcPositiveRatioMeasure	0.00	1.00	0.00	Measure for the ration of light reflected (versus absorbed) by this surface. Value of 0.00 means the value has not been set.
			SurfaceColor		STRING	n/a	n/a	empty string	Color of this surface, using the xxx color standard
			SurfaceTexture		STRING	n/a	n/a	empty string	Surface bumpiness - using the xxx standard
			ThermalCoefficient		REAL	0.00	see type	0.00	Thermal "U-value" per unit thickness. Value of 0.00 means the value has not been set.
IfcBuildingElement (assembly using IfcRelAssembles)	6	**IfcRoofFrame**							A collection (assembly) of structural frame elements supporting a roof or segments of a roof
		I_RoofFrame : GenericType, SupportsRoofSlabs, DesignLoads, FireRating							
			GenericType		IfcRoofFrameTypeEnum	Framed	Pneumatic	Trussed	Predefined generic types are specified in an Enum. Type driven Psets are defined for each generic type (as the required attributes differ). The GenericType for a given instance drives determines the type of Pset attached at runtime throug the associated TypeDef object (defined at the IfcObject supertype).
			SupportsRoofSlabs		LIST [0:?] OF Ref [IfcRoofslab]	n/a	n/a	empty list	List of relationships - to the Roofslab objects supported by this RoofFrame
			DesignLoads		IfcOccurrencePropertySet (Pset_StructuralAreaLoads)	n/a	n/a	NIL	Reference to a Pset in the ExtensionPsets for this object -- the one defining the design loads for this roof frame. Pset_StructuralAreaLoads defines both the "Live" and "Dead" loads.
			FireRating		IfcTimeDurationMeasure	see type	see type	0.00	Time measure for which this assembly is rated in case of fire. Value of 0.00 means this attribute not set.

IfcRelationship	7	**IfcRelJoinsElements**							Expansion joint, edge condition, control joint.
			I_RelJoinsElements : RelatingObject, RelatedObjects, JointType, RangeOfMovement, ConstructionDetails, SpecSection, WaterProofing, FireRating, VentilationRequired, ManufactureInfo, ObjectLifeCycle						
			RelatingObject		Ref [IfcBuildingElement]	n/a	n/a	NIL	Primary object at the joint defined by this relationship
			RelatedObjects		LIST [1:?] OF Ref [IfcBuildingElement]	n/a	n/a	empty list	Secondard objects joined at the joint defined by this relationship
			JointElements		LIST [0:?] OF Ref [IfcBuildingElement]	n/a	n/a	empty list	Objects that make up the joint (fill the gap)
			JoinType		IfcJointTypeEnum	control	expansion	control	Purpose of joint
			RangeOfMovement		IfcPositiveLengthMeasure	0.00	n	0.00	Distance the joint can open before failing
			ConstructionDetails		LIST [0:?] OF Ref [IfcDocumentReference]	na	na	na	List of references to drawing documents which define construction details
			SpecSection		IfcDocumentReference	na	na	NIL	Reference to a section of the specification
			WaterProofing		BOOLEAN	FALSE	TRUE	FALSE	flag that indicates that the joint should be waterproof or not
			FireRating		IfcTimeDurationMeasure	see type	see type	60 min	Time duration for fire resistence the roof assembly is rated
			VentilationRequired		BOOLEAN	FALSE	TRUE	FALSE	Is ventilation required for this joint?
			ManuafactureInfo		IfcOccurrencePropertySet (Pset_ManufactureInfo)	n/a	n/a	NIL	Reference to information about the manufacturer of this joint assemlby (if any). ID of Pset_ManufactureInfo attached to object in the Extension Psets list
			ObjectLifeCycle		IfcOccurrencePropertySet (Pset_ObjectLifeCycle)	na	na	na	Reference to ObjectLifeCycle Pset - attached to object in the Extension Pset list
IfcBuildingElement	8	**IfcScreen**							
			I_Screen : GenericType						
			GenericType		IfcScreenTypeEnum	ScreenAssembly	ScreenDoorOrGate	ScreenPanel	Predefined generic types are specified in an Enum. Type driven Psets are defined for each generic type (as the required attributes differ). The GenericType for a given instance drives determines the type of Pset attached at runtime throug the associated TypeDef object (defined at the IfcObject supertype).

6.1.1.6. Componentization of Doors/Windows

		Object Type Name	Interface name	OPT INV DER					
Subtype of			Attribute / Relation name		Data Type/Related Object Type	Min	Max	Default	Definition
IfcBuildingElement	19	**IfcDoorPanel**							
			I_DoorPanel : GenericType, PanelHeight, PanelWidth, PanelThickness						
			GenericType		IfcDoorPanelTypeEnum	SwingDoor	RollupDoor	SwingDoor	Predefined generic types are specified in an Enum. Type driven Psets are defined for each generic type (as the required attributes differ). The GenericType for a given instance drives determines the type of Pset attached at runtime throug the associated TypeDef object (defined at the IfcObject supertype).
			PanelHeight		IfcPositiveLengthMeasure	0	see type	1800	Overall Height of this panel. Note this can be derived from the 'Shape' and included for convenience use by applications that cannot derive this from the shape.

			PanelWidth	IfcPositiveLengthMeasure	0	see type	900	Overall Width of this panel. Note this can be derived from the 'Shape' and included for convenience use by applications that cannot derive this from the shape.
			PanelThickness	IfcPositiveLengthMeasure	0	see type	50	Overall Thickness of this panel. Note this can be derived from the 'Shape' and included for convenience use by applications that cannot derive this from the shape.
IfcBuildingElement	20	**IfcWindowPanel**						
		I_WindowPanel : GenericType, PanelHeight, PanelWidth, PanelDepth, PanelFrameThickness, StileThickness, StileDepth						
			GenericType	IfcWindowPanelTypeEnum	FixedPanel	SwingPanel	FixedPanel	Predefined generic types are specified in an Enum. Type driven Psets are defined for each generic type (as the required attributes differ). The GenericType for a given instance drives determines the type of Pset attached at runtime throug the associated TypeDef object (defined at the IfcObject supertype).
			PanelHeight	IfcPositiveLengthMeasure	0	see type	1800	Overall Height of this panel. Note this can be derived from the 'Shape' and included for convenience use by applications that cannot derive this from the shape.
			PanelWidth	IfcPositiveLengthMeasure	0	see type	900	Overall Width of this panel. Note this can be derived from the 'Shape' and included for convenience use by applications that cannot derive this from the shape.
			PanelDepth	IfcPositiveLengthMeasure	0	see type	50	Overall Depth of this panel. Note this can be derived from the 'Shape' and included for convenience use by applications that cannot derive this from the shape.
			PanelFrameThickness	IfcPositiveLengthMeasure	0	see type	25	Thickness (width in plane parallel to glazing) of the panel frame. Note the PanelFrameDepth is taken to be = Depth for the panel
			StileThickness	IfcPositiveLengthMeasure	0	see type	25	Thickness (width in plane parallel to glazing) of the stiles dividing any glass panes
			StileDepth	IfcPositiveLengthMeasure	0	see type	25	Depth (dimension in plane perpendicular to glazing) of the stiles dividing any glass panes
IfcBuildingElement	21	**IfcDoorOrWindowFrame**						
		I_DoorORWindowFrame : GenericType, calc_FrameDepth, calc_FrameThickness						
			GenericType	IfcDoorOrWindowFrameTypeEnum	FramedInPlace	PreHung	FramedInPlace	Predefined generic types are specified in an Enum. Type driven Psets are defined for each generic type (as the required attributes differ). The GenericType for a given instance drives determines the type of Pset attached at runtime throug the associated TypeDef object (defined at the IfcObject supertype).
			calc_FrameDepth	IfcPositiveLengthMeasure	0	see type	150	Depth of the frame horizontal section, from front face (facing space 'A') to back face (facing space 'B')
			calc_FrameThickness	IfcPositiveLengthMeasure	0	see type	50	Thickness of the frame horizontal section, measured from inside of frame (at door panel) to outside of frame (at the rought opening in the host wall)
	22	**IfcGlazing**						
		I_Glazing : GenericType						
			GenericType	IfcGlazingTypeEnum	FramedInPlace	PreHung	FramedInPlace	Predefined generic types are specified in an Enum. Type driven Psets are defined for each generic type (as the required attributes differ). The GenericType for a given instance drives determines the type of Pset attached at runtime throug the associated TypeDef object (defined at the IfcObject supertype).

6.1.2. Type Definitions

#	TypeDef Name	Class being Typed	Generic Type / Description or definition	Specific Type	Set #	Shared Pset	Set #	Occurrence Pset
1	**BeamType**		Supports the definition of standard Beam types and associated PropertySets					
		IfcBeam		Common Pset			#N/A	Pset_BeamCommon
			StairStringer					
				< Any value >			4	Pset_BeamStairStringer
2	**BuiltInAccessoryType**		Supports the definition of standard Accessory types and associated PropertySets					
		IfcBuiltInAccessory		Common Pset			1	Pset_AccessoryCommon
			Bathroom					
				< Any value >			2	Pset_AccessoryBathroom
			DoorOrWindowHardware		3	Pset_AccessoryDoorOrWindowHardware		
3	**CabinetType**		Supports the definition of standard Cabinet types and associated PropertySets					
		IfcCabinet		Common Pset			5	Pset_CabinetCommon
			Bathroom					
				< Any value >	6	Pset_CabinetBathroom		
			Kitchen					
				< Any value >	7	Pset_CabinetKitchen		
			Storage					
				< Any value >	8	Pset_CabinetStorage		
			Laundry					
				< Any value >	9	Pset_CabinetLaundry		
			Office					
				< Any value >	10	Pset_CabinetOffice		
4	**CounterType**		Supports the definition of standard Counter types and associated PropertySets					
		IfcCounterOrShelf		Common Pset			11	Pset_CounterOrShelfCommon
			Counter					
				< Any value >			12	Pset_Counter
			Shelf					
				< Any value >			47	Pset_Shelf
5	**CoveringType**		Supports the definition of standard Covering types and associated PropertySets					
		IfcCovering		Common Pset			#N/A	Pset_CoveringCommon
			CoveringMillwork					
				< Any value >			13	Pset_CoveringMillwork

6	**CurtainWallElementType**		Supports the definition of standard CurtainWallElement types and associated PropertySets					
		IfcCurtainWallElement		Common Pset			14	Pset_CurtainWallElementCommon
			Clad Panels					
				< Any value >	15	Pset_CurtainWallCladPanel		
			Glazing Panel					
				< Any value >	16	Pset_CurtainWallGlazingPanel		
			Ornamental Projection					
				< Any value >	17	Pset_CurtainWallProjectionOrnamental		
			Shade or Overhang					
					18	Pset_CurtainWallShadeOverhang		
			Spandrel Panel					
					19	Pset_CurtainWallSpandrelPanel		
7	**DistributionElementType**		Supports the definition of standard DistributionElement types and associated PropertySets					
		IfcDistributionElement		Common Pset			#N/A	Pset_DistributionElementCommon
			Roof Drainage					
				Drain			21	Pset_DistributionDrain
				Gutter			22	Pset_DistributionGutter
				Scupper			20	Pset_DistributionScupper
8	**DocumentReferenceType**		Supports the definition of standard DocumentReference types and associated PropertySets					
		IfcDocumentReference		Common Pset			#N/A	Pset_DocumenReferenceCommon
			Specifications					
				< Any value >			23	Pset_DocumentSpecSection
9	**EquipmentType**		Supports the definition of standard Equipment types and associated PropertySets					
		IfcEquipment		Common Pset			#N/A	Pset_EquipmentCommon
			Elevator					
				< Any value >			25	Pset_EquipmentElevator
			Escalator					
				< Any value >			26	Pset_EquipmentEscalator
			Window Cleaning					
				< Any value >			27	Pset_EquipmentWindowCleaningElement
10	**PathwayType**		Supports the definition of standard Pathway types and associated PropertySets					
		IfcPathway		Common Pset			#N/A	Pset_PathwayCommon
			RoofAccessPath					
				< Any value >			30	Pset_PathwayRoofAccessPath

11	**PermeableOpeningCoverType**		Supports the definition of standard PermeableOpeningCover types and associated PropertySets				
		IfcPermeableOpeningCover		Common Pset		34	Pset_PermOpenCoverCommon
			Grill				
				< Any value >		31	Pset_PermOpenCoverGrill
			Louver				
				< Any value >		32	Pset_PermOpenCoverLouver
			Screen				
				< Any value >		33	Pset_PermOpenCoverScreen
12	**PlumbingFixtureType**		Supports the definition of standard PlumbingFixture types and associated PropertySets				
		IfcPlumbingFixture		Common Pset		35	Pset_PlumbingFixtureCommon
			Faucet				
				< Any value >		#N/A	Pset_PlumbingFixtureFaucet
			Sink				
				< Any value >		#N/A	Pset_PlumbingFixtureSink
			Shower				
				< Any value >		#N/A	Pset_PlumbingFixtureShower
			Toilet				
				< Any value >		#N/A	Pset_PlumbingFixtureToilet
			Urinal				
				< Any value >		#N/A	Pset_PlumbingFixtureUrinal
13	**VisualScreenType**		Supports the definition of standard Screen types and associated PropertySets				
		IfcVisualScreen		Common Pset		42	Pset_VisualScreenCommon
			Assembly				
				< Any value >		43	Pset_VisualScreenAssembly
			DoorOrGate				
				Restroom Partition Door		41	Pset_VisualScreenRestroomPartitionDoor
				< Any other value >		46	Pset_VisualScreenDoorOrGate
			Panel				
				Restroom Partition		40	Pset_VisualScreenRestroomPartition
				< Any other value >		45	Pset_VisualScreenPanel
			Post				
				< Any value >		44	Pset_VisualScreenPost
10	**RailingType**		Supports the definition of standard Railing types and associated PropertySets				
		IfcRailing		Common Pset		36	Pset_RailingCommon
			Handrail				
				< Any value >		37	Pset_RailingHandrail
			Guardrail				

			< Any value >	38	Pset_RailingGuardrail
			Balustrade		
			< Any value >	39	Pset_RailingBalustrade
11	**SpaceType**		Supports the definition of standard Space types and associated PropertySets		
		IfcSpace	Common Pset	#N/A	Pset_SpaceCommon
			CirculationSpace		
			StairShaft	49	Pset_SpaceStairShaft
			< Any other value >	#N/A	Pset_SpaceCirculation
			TechnicalSpace		
			Elevator Shaft	48	Pset_SpaceElevatorShaft
			< Any other value >	#N/A	Pset_SpaceTechnical
12	**WallType**		Supports the definition of standard Wall types and associated PropertySets		
		IfcWall	Common Pset	#N/A	Pset_WallCommon
			Partition		
			< Any value >	#N/A	Pset_WallPartition
13	**WindowType**		Supports the definition of standard Window types and associated PropertySets		
		IfcWindow	Common Pset	#N/A	Pset_WindowCommon
			Skylight		
			< Any value >	51	Pset_WindowSkylight
14	**Extension Psets**		PropertySets which extend the definition of many types of objects		
		Any Object			
			Maintenance	24	Pset_ElementMaintenance
			Maintenance Record	28	Pset_MaintenanceRecord
			Object Life Cycle	29	Pset_ObjectLifeCycle

6.1.3. Property Sets

	PropertySet (Pset) Name Attribute / Relation name	Definition	Data Type or Related Object	Min	Max	Default
1	**Pset_AccessoryCommon**					
	ManuafactureInfo	reference to Manufacturer information	ObjectReference (Pset_ManufactureInfo)	n/a	n/a	NIL
	ManufacturerMaterial	Material selection - from the manufacturer's material options for this fixture type	IfcString	n/a	n/a	empty string
	ManufacturerColor	Color selection - from the manufacturer's color options for this fixture type	IfcString	n/a	n/a	empty string
	ManufacturerFinish	Finish selection - from the manufacturer's finish options for this fixture type	IfcString	n/a	n/a	empty string
	Target objects for this PropertySet					
	IfcBuiltInAccessory					
2	**Pset_AccessoryBathroom**	These are what are commonly referred to as "Bathroom Accessories"				

	CommonAccessoryProperties	Reference to the SharedPropertySet (Pset_AccessoryCommon). Contains the shared values for this type -- of properties that are stored for all Screen elements.	IfcObjectReference (Pset_AccessoryCommon)	n/a	n/a	NIL
	Property2	Not yet defined				
	Target objects for this PropertySet					
	IfcBuiltInAccessory					
3	**Pset_AccessoryDoorOrWindowHardware**					
	CommonAccessoryProperties	Reference to the SharedPropertySet (Pset_AccessoryCommon). Contains the shared values for this type -- of properties that are stored for all Screen elements.	IfcObjectReference (Pset_AccessoryCommon)	n/a	n/a	NIL
	ProjectHwGroupReference	Project reference ID for this standard collection of hardware elements for doors	IfcString	see type	see type	empty string
	TypeDescription	Description for this type of frame (note name is captured in the TypeDef object that references this PropertySet)	IfcString	see type	see type	empty string
	DoorHardwareElementIndexList	A list of indicies into the enumeration referenced by DoorHardwareElementEnum. Note: this list will be implemented as a shared Pset_DoorHardwareGroupElements - a list of IfcInteger indicies into that enum.	LIST [1:?] OF IfcInteger	1	10	1
	DoorHardwareElementEnum	Reference to a Pset enumerating all possible door hardware elements for this hardware group. Note: this will be implemented as a shared PropertySet (Pset_DoorHardwareElementEnum) of IfcString (values enumerated at right).	IfcObjectReference (Pset_DoorHardwareElementEnum - ENUMERATION OF (Hingeset, Lockset, Handset, Deadbolt, Kickplate, Pushplate, Peephole, Knocker, DoorStop, Passthrough))	n/a	n/a	NIL
	Target objects for this PropertySet					
	IfcBuiltInAccessory					
4	**Pset_BeamStairStringer**					
	Slope	Slope for this stringer - relative to horizontal (0.0 degrees).	IfcAngleMeasure	0.0	see type	0.0
	ConstructionDetails	List of references to construction detail drawings	LIST [0:?] OF ObjectReference (IfcDocumentReference)	n/a	n/a	empty list
	Target objects for this PropertySet					
	IfcBeam					
5	**Pset_CabinetCommon**					
	ManuafactureInfo	reference to Manufacturer information	ObjectReference (Pset_ManufactureInfo)	n/a	n/a	NIL
	ConstructionDetail	Reference to construction detail drawing	Ref [IfcDocumentReference]	n/a	n/a	NIL
	SpecSection	Reference to a section of the construction specification	ObjectReference (IfcDocumentReference)	n/a	n/a	NIL
	Target objects for this PropertySet					
	IfcCabinet					
6	**Pset_CabinetBathroom**					
	CommonCabinetProperties	Reference to the SharedPropertySet (Pset_CabinetCommon). Contains the shared values for this type -- of properties that are stored for all types of cabinets.	IfcObjectReference (Pset_CabinetCommon)	n/a	n/a	NIL
	Property2	Not yet defined				
	Target objects for this PropertySet					
	IfcCabinet					

7	**Pset_CabinetKitchen**					
	CommonCabinetProperties	Reference to the SharedPropertySet (Pset_CabinetCommon). Contains the shared values for this type -- of properties that are stored for all types of cabinets.	IfcObjectReference (Pset_CabinetCommon)	n/a	n/a	NIL
	Property2	Not yet defined				
	Target objects for this PropertySet					
	IfcCabinet					
8	**Pset_CabinetStorage**					
	CommonCabinetProperties	Reference to the SharedPropertySet (Pset_CabinetCommon). Contains the shared values for this type -- of properties that are stored for all types of cabinets.	IfcObjectReference (Pset_CabinetCommon)	n/a	n/a	NIL
	Property2	Not yet defined				
	Target objects for this PropertySet					
	IfcCabinet					
9	**Pset_CabinetLaundry**					
	CommonCabinetProperties	Reference to the SharedPropertySet (Pset_CabinetCommon). Contains the shared values for this type -- of properties that are stored for all types of cabinets.	IfcObjectReference (Pset_CabinetCommon)	n/a	n/a	NIL
	Property2	Not yet defined				
	Target objects for this PropertySet					
	IfcCabinet					
10	**Pset_CabinetOffice**					
	CommonCabinetProperties	Reference to the SharedPropertySet (Pset_CabinetCommon). Contains the shared values for this type -- of properties that are stored for all types of cabinets.	IfcObjectReference (Pset_CabinetCommon)	n/a	n/a	NIL
	Property2	Not yet defined				
	Target objects for this PropertySet					
	IfcCabinet					
11	**Pset_CounterOrShelfCommon**					
	ManuafactureInfo	reference to Manufacturer information	ObjectReference (Pset_ManufactureInfo)	n/a	n/a	NIL
	ConstructionDetail	Reference to construction detail drawing	Ref [IfcDocumentReference]	n/a	n/a	NIL
	SpecSection	Reference to a section of the construction specification	ObjectReference (IfcDocumentReference)	n/a	n/a	NIL
	Target objects for this PropertySet					
	IfcCounterOrShelf					
12	**Pset_Counter**					
	CommonCounterOrShelfProperties	Reference to the SharedPropertySet (Pset_CounterOrShelfCommon). Contains the shared values for this type -- of properties that are stored for all types of counters and shelves.	IfcObjectReference (Pset_CounterOrShelfCommon)	n/a	n/a	NIL
	Property2	Not yet defined				
	Target objects for this PropertySet					
	IfcCounterOrShelf					
13	**Pset_CoveringMillwork**					
	CommonCoveringProperties	Reference to a SharedPropertySet (Pset_CoveringCommon). Contains the shared values for this type -- of properties that are stored for all Covering elements.	IfcObjectReference (Pset_CoveringCommon)	n/a	n/a	NIL

	ConstructionDetail	Reference to a construction detail drawing file	ObjectReference (IfcDocumentReference)	n/a	n/a	NIL
	Target objects for this PropertySet					
	IfcCovering					
14	**Pset_CurtainWallElementCommon**	Defines properties common for all CurtainWall elements.				
	ManufactureInformation	Reference to a SharedPropertySet - Pset_ManufactureInformation, which defines information about the manufacture of this element.	IfcObjectReference (Pset_ManufactureInformation)	n/a	n/a	NIL
	LifecycleInformation	Reference to lifecycle	IfcObjectReference (Pset_ObjectLifecycle)	na	na	NIL
	ConstructionDetails	List of references to a detail drawings	LIST [0:?] OF IfcDocumentReference	n/a	n/a	empty list
	SpecSection	Document reference to specification section	IfcDocumentReference	n/a	n/a	NIL
	BldgCodeRefs	List of document references to building codes	LIST [0:?] OF IfcDocumentReference	n/a	n/a	empty list
	Target objects for this PropertySet					
	IfcCurtainWallElement					
15	**Pset_CurtainWallCladPanel**					
	CommonCurtainWallElementProperties	Reference to the 'parent' SharedPropertySet (Pset_CurtainWallElementType). Contains the shared values for this type -- of properties that are stored for all CurtainWallElements.	IfcObjectReference (Pset_CurtainWallElementType)	n/a	n/a	NIL
	Target objects for this PropertySet					
	IfcCurtainWallElement					
16	**Pset_CurtainWallGlazingPanel**					
	CommonCurtainWallElementProperties	Reference to the 'parent' SharedPropertySet (Pset_CurtainWallElementType). Contains the shared values for this type -- of properties that are stored for all CurtainWallElements.	IfcObjectReference (Pset_CurtainWallElementType)	n/a	n/a	NIL
	Target objects for this PropertySet					
	IfcCurtainWallElement					
17	**Pset_CurtainWallProjectionOrnamental**					
	CommonCurtainWallElementProperties	Reference to the 'parent' SharedPropertySet (Pset_CurtainWallElementType). Contains the shared values for this type -- of properties that are stored for all CurtainWallElements.	IfcObjectReference (Pset_CurtainWallElementType)	n/a	n/a	NIL
	Description	Description of this projecting element	IfcString	n/a	n/a	empty string
	Weight	Total weight of projection	IfcMassMeasure	0.0	see type	0.0
	ConstructionDetails	Reference to detail construction drawings for connection to façade (ie. bolt, screw or fastener detail)	LIST [0:?] OF ObjectReference (IfcDocumentReference)	n/a	n/a	NIL
	Services	References to building services needed (e.g. electrical to an operable canopy)	LIST [0:?] OF ObjectReference (IfcSystem)	n/a	n/a	empty list
	Target objects for this PropertySet					
	IfcCurtainWallElement					
18	**Pset_CurtainWallShadeOverhang**					
	CommonCurtainWallElementProperties	Reference to the 'parent' SharedPropertySet (Pset_CurtainWallElementType). Contains the shared values for this type -- of properties that are stored for all CurtainWallElements.	IfcObjectReference (Pset_CurtainWallElementType)	n/a	n/a	NIL
	Target objects for this PropertySet					
	IfcCurtainWallElement					

19	**Pset_CurtainWallSpandrelPanel**					
	CommonCurtainWallElementProperties	Reference to the 'parent' SharedPropertySet (Pset_CurtainWallElementType). Contains the shared values for this type -- of properties that are stored for all CurtainWallElements.	IfcObjectReference (Pset_CurtainWallElementType)	n/a	n/a	NIL
	Target objects for this PropertySet					
	IfcCurtainWallElement					
20	**Pset_DistributionScupper**	General opening/edge condition designed to distribute (convey) overflow drainage from a roof or deck.				
	ManuafactureInfo	reference to Manufacturer information	ObjectReference (Pset_ManufactureInfo)	n/a	n/a	NIL
	ConstructionDetail	Reference to a construction detail drawing	ObjectReference [IfcDocumentReference]	na	na	na
	SpecSection	Reference to a section of the specification	ObjectReference (IfcDocumentReference)	na	na	NIL
	Target objects for this PropertySet					
	IfcDistributionElement	Roof or Downspout				
21	**Pset_DistributionDrain**					
	ManuafactureInfo	reference to Manufacturer information	ObjectReference (Pset_ManufactureInfo)	n/a	n/a	NIL
	TributaryAreaDrained	Area for this this is the primary drain. Value of 0.00 means this value has not been set.	IfcAreaMeasure	see type	see type	0.00
	FlowCapacity	Calculated capacity of drain flow. Value of 0.00 means this value has not been set.	IfcFlowMeasure	see type	see type	0.00
	ConstructionDetail	Reference to a construction detail drawing	ObjectReference (IfcDocumentReference)	na	na	NIL
	SpecSection	Reference to a section of the specification	ObjectReference (IfcDocumentReference)	na	na	NIL
	Target objects for this PropertySet					
	IfcDistributionElement					
22	**Pset_DistributionGutter**					
	Slope	angle of the gutter to allow for drainage	IfcAngleMeasure	0.0	see type	0.0
	FlowCapacity	Calculated capacity of drain flow. Value of 0.00 means this value has not been set.	IfcFlowMeasure	see type	see type	0.00
	ConstructionDetail	Reference to a construction detail drawing	ObjectReference (IfcDocumentReference)	na	na	NIL
	SpecSection	Reference to a section of the specification	ObjectReference (IfcDocumentReference)	na	na	NIL
	Target objects for this PropertySet					
	IfcDistributionElement					
23	**Pset_DocumentSpecSection**					
	CommonDocumentReferenceProperties	Reference to a SharedPropertySet (Pset_DocumentReferenceCommon), which contains the properties that are stored for all types of Document References.	IfcObjectReference (Pset_DocumentReferenceCommon)	n/a	n/a	NIL
	SectionID	Section number or ID for the referenced section	IfcString	n/a	n/a	empty string
	OffsetToSection	Byte count offset from beginning of file to the beginning of the referenced section. Value of 0 means offset not set.	IfcInteger	0	see type	0
	Target objects for this PropertySet					
	IfcDocumentReference					
24	**Pset_ElementMaintenance**					

				ElementMaintenanceConditionEnum	Reference to nested enumeration property set Pset_ElementMaintenanceConditionEnum. This enumeration defines the general conditions for a building element requiring routine maintenance.	ENUMERATION OF (GoodCondition, RequiresMonitoring, RequiresRoutineMaintenance, RequiresRepair, RequiresReplacement, Other, NotKnown, Unset)	n/a	n/a	NIL
				ElementMaintenanceConditionIndex	Index into the nested enumeration property set Pset_ElementMaintenanceConditionEnum	IfcInteger	1	N	1
				ServiceActor	The person or maintenance service provider responsible for the maintenance of the element	IfcActorSelect	n/a	n/a	NIL
				MaintenanceRecords	List of references to maintenance records for this element.	LIST [0:?] OF IfcSharedPropertySet (Pset_MaintenanceRecord)	n/a	n/a	empty list
			Target objects for this PropertySet						
				IfcElement					
25		**Pset_EquipmentElevator**							
				CommonEquipmentProperties	Reference to a SharedPropertySet (Pset_EquipmentCommon) which defines properties that are stored for all types of equipment.	IfcSharedPropertySet (Pset_EquipmentCommon)	n/a	n/a	NIL
				Occupancy	Number of occupants	IfcInteger	0	see type	0
				ManufactureInfo	Nested Pset defining manufacturing info	IfcSharedPropertySet (Pset_ManufacturInfo)	n/a	n/a	NIL
				LoadCapacity	Weight capacity of elevator	IfcMassMeasure	see type	see type	0
				ClientBrief	Reference to program to gain requirements for occupancy	ObjectReference (IfcClientBrief)	n/a	n/a	NIL
			Target objects for this PropertySet						
				IfcEquipment					
26		**Pset_EquipmentEscalator**							
				CommonEquipmentProperties	Reference to a SharedPropertySet (Pset_EquipmentCommon) which defines properties that are stored for all types of equipment.	IfcSharedPropertySet (Pset_EquipmentCommon)	n/a	n/a	NIL
				Capacity	number of people that can be moved from the top to the bottom	IfcInteger	0	see type	0
				ManufactureInfo	reference to Pset_ManufactureInfo	ObjectReference (Pset_ManufacturInfo)	n/a	n/a	NIL
				ClientBrief	Link to program to gain requirements for occupancy	ObjectReference (IfcClientBrief)	n/a	n/a	NIL
			Target objects for this PropertySet						
				IfcEquipment					
27		**Pset_EquipmentWindowCleaningElement**							
				CommonEquipmentProperties	Reference to a SharedPropertySet (Pset_EquipmentCommon) which defines properties that are stored for all types of equipment.	IfcSharedPropertySet (Pset_EquipmentCommon)	n/a	n/a	NIL
				WindowCleaningElementTypeEnum	Enumeration of the various	Enum (Apparatus, Carriage, Rails, Rigging, Tracks)	Apparatus	Tracks	Carriage
				WindowCleaningElementTypeIndex	Index (in the enum above) indicating the type of window cleaning system element for this object	IfcInteger	1	5	2
			Target objects for this PropertySet						
				IfcEquipment					
28		**Pset_MaintenanceRecord**							
				MaintenanceDate	Date maintenance performed	IfcDateTimeSelect	see type	see type	1-Jan-72
				MaintenanceReason	Description of Problem	IfcString	see type	see type	empty string
				MaintenanceDescription	Description of what work was performed	IfcString	see type	see type	empty string

	Crew	Maintenance crew involved	LIST [0:?] OF IfcActorSelect	n/a	n/a	empty list
	Target objects for this PropertySet					
	IfcRoof, Pset_LifeCycle					
29	**Pset_ObjectLifeCycle**					
	InstallationDate	Date of installation	IfcDateTimeSelect	see type	see type	1-Jan-72
	ServiceLife	Time period inwhich the object is projected to last without replacement	IfcTimeDurationMeasure	see type	see type	0
	Warranty	Legal description of time period that the manufacturer is responsible for replacement	IfcTimeDurationMeasure	see type	see type	0
	MaintenanceInterval	Time period between each maintenance cycle	IfcTimeDurationMeasure	see type	see type	0
	MaintenanceRequirements	Requirments for maintenance	IfcString	see type	see type	empty string
	MaintenanceHistory	List of links to maintenance records	LIST [0:?] IfcObjectReference (Pset_MaintenanceRecord)	n/a	n/a	empty list
	LifeCycleCost	Total cost of object (may be an assembly) over the LifeCycleCostPeriod	IfcCost	0.0	see type	0.0
	LifeCycleCostPeriod	Life for which the LifeCycleCost has been calculated	IfcTimeDurationMeasure	see type	see type	0.0
	SalvageValue	Value if recycled or returned when replacement	IfcCost	0.0	n	0.0
	Target objects for this PropertySet					
	IfcAssemblyCurtainWall, IfcRoofSlab					
30	**Pset_PathwayRoofAccessPath**					
	PathLength	Description of walk to mechanical	IfcPositiveLengthMeasure	0.0	see type	0.0
	PathWidth	Distance across path	IfcPositiveLengthMeasure	0.0	see type	0.0
	Target objects for this PropertySet					
	IfcPathway					
31	**Pset_PermOpenCoverGrill**					
	CommonPermeableOpeningCover Properties	Reference to the 'parent' SharedPropertySet (Pset_PermeableOpeningCoverType). Contains the shared values for this type -- of properties that are stored for all PermeableOpeningCovers.	IfcObjectReference (Pset_PermeableOpeningC overType)	n/a	n/a	NIL
	GrillMaterial	Primary material from which this grill is made - an index into the MaterialSet associated with this building element	IfcInteger	0	see type	0
	HorzSpacing	Spacing of the screening wire at the angle set by Orientation. "0.0" indicates value not set.	IfcPositiveLengthMeasure	see type	see type	0.0
	VertSpacing	Spacing of the screening wire at the angle perpendicular to that set by Orientation. "0.0" indicates value not set.	IfcPositiveLengthMeasure	see type	see type	0.0
	FinWidth	Width (when viewed from finished side) of the fins in this grill. Value of 0.0 means value not set.	IfcPositiveLengthMeasure	see type	see type	0.0
	FinDepth	depth (finished side to back side) of the fins in this grill. Value of 0.0 means value not set.	IfcPositiveLengthMeasure	see type	see type	0.0
	Target objects for this PropertySet					
	IfcPermeableOpeningCover					
32	**Pset_PermOpenCoverLouver**	A type of permeable cover for an opening (which allows airflow). Louvers may be placed in any Opening.				
	CommonPermeableOpeningCover Properties	Reference to the 'parent' SharedPropertySet (Pset_PermeableOpeningCoverType). Contains the shared values for this type -- of properties that are stored for all PermeableOpeningCovers.	IfcObjectReference (Pset_PermeableOpeningC overType)	n/a	n/a	NIL
	LouverMaterial	Primary material from which this louver is made - an index into the MaterialSet associated with this building element	IfcInteger	0	see type	0

				FinSpacing	Distance between adjacent fins. "0.0" indicates value not set.	IfcPositiveLengthMeasure	see type	see type	0.0
				FinAngle	Slope angle of the fins, in cross-sectional view with finished (or exterior) face on the right side of the section. Horzontal fin angle is taken to be zero ("0") angle.	IfcAngleMeasure	0.0	<360.0	0.0
				FinDepth	Fin depth measure, in cross-sectional view. "0.0" indicates value not set.	IfcPositiveLengthMeasure	see type	see type	0.0
				InsideScreen	Reference to a screen on the inside of these louvers	ObjectReference (Pset_PermOpenCoverScreen)	n/a	n/a	NIL
			Target objects for this PropertySet						
				IfcPermeableOpeningCover					
33		**Pset_PermOpenCoverScreen**							
				CommonPermeableOpeningCoverProperties	Reference to the 'parent' SharedPropertySet (Pset_PermeableOpeningCoverType). Contains the shared values for this type -- of properties that are stored for all PermeableOpeningCovers.	IfcObjectReference (Pset_PermeableOpeningCoverType)	n/a	n/a	NIL
				ScreenMaterial	Primary material from which this screen is made - an index into the MaterialSet associated with this building element	IfcInteger	0	see type	0
				HorzSpacing	Spacing of the screening wire at the angle set by Orientation. "0.0" indicates value not set.	IfcPositiveLengthMeasure	see type	see type	0.0
				VertSpacing	Spacing of the screening wire at the angle perpendicular to that set by Orientation. "0.0" indicates value not set.	IfcPositiveLengthMeasure	see type	see type	0.0
				ScreenThickness	Thickness of the screening wire	IfcPositiveLengthMeasure	see type	see type	0.5
			Target objects for this PropertySet						
				IfcPermeableOpeningCover					
34		**Pset_PermOpenCoverCommon**			Permeable cover for an opening which allows airflow (definition BS 6100)				
				TypeDescription	Description for this type of louver (note name is captured in the TypeDef object that references this PropertySet)	IfcString	see type	see type	empty string
				ManufactureInformation	Reference to a SharedPropertySet - Pset_ManufactureInformation, which defines information about the manufacture of this door hardware.	IfcObjectReference (Pset_ManufactureInformation)	n/a	n/a	NIL
				RequiredOpeningHeight	Overall Height of the required opening for this louver. Note this can be derived from the 'ProductShape' and is included for convenience use by applications that cannot derive this from the shape. Zero means this property has not been set.	IfcPositiveLengthMeasure	0	see type	0
				RequiredOpeningWidth	Overall Width of the required opening for this louver. Note this can be derived from the 'ProductShape' and is included for convenience use by applications that cannot derive this from the shape. Zero means this property has not been set.	IfcPositiveLengthMeasure	0	see type	0
				FrameWidth	Average length measure, when viewed from the finished face, from the edge of the louver to fins.	IfcPositiveLengthMeasure	see type	see type	1.0
				FrameDepth	Measure of the frame depth (front to back)	IfcPositiveLengthMeasure	see type	see type	1.0
				Orientation	Orientation angle, when facing the finished side of installed louvers. Horzontal is taken to be zero ("0") angle. Angle is positive in counter-clockwise rotation.	IfcAngleMeasure	0.0	<360.0	0.0
				ConstructionDetail	Reference to a construction detail drawing	IfcObjectReference (IfcDocumentReference)	see type	see type	NIL
				SpecSection	Reference to a section in the construction specifications	IfcObjectReference (IfcDocumentReference)	see type	see type	NIL
				FreeAreaVentilation	Actual usable Area. Zero means this value has not been set.	IfcAreaMeasure	0	see type	0
				ClearanceSpace	Distance needed for correct operation/air flow	ObjectReference (IfcSpace)	see type	see type	NIL
				Operable	Designation of operability of this cover	IfcBoolean	FALSE	TRUE	FALSE
				Control	Reference to control system if needed	ObjectReference (IfcSystem)	see type	see type	NIL

	Target objects for this PropertySet					
	IfcPermeableOpeningCover					
35	**Pset_PlumbingFixtureCommon**					
	ManuafactureInfo	reference to Manufacturer information	ObjectReference (Pset_ManufactureInfo)	n/a	n/a	NIL
	FunctionalHeight	Height from floor to functional opening. Value of 0.0 means this property not set.	IfcPositiveLengthMeasure	0.0	see type	0.0
	MountingHeight	height at which the item gets connect to the wall. Value of 0.0 means this property not set.	IfcPositiveLengthMeasure	0.0	see type	0.0
	MountingType	Description of the method for mounting	IfcString	n/a	n/a	empty string
	DrainConnectPoint	Reference to the connection object relating this plumbing fixture to the sewer piping system (the drain)	ObjectReference (IfcRelConnectsElements)	n/a	n/a	NIL
	HwconnectPoint	Reference to the connection object relating this plumbing fixture to the hot water plumbing system.	ObjectReference (IfcRelConnectsElements)	n/a	n/a	NIL
	CWconnectPoint	Reference to the connection object relating this plumbing fixture to the cold water plumbing system	ObjectReference (IfcRelConnectsElements)	n/a	n/a	NIL
	ElectricalConnectPoint	Reference to the connection object relating this plumbing fixture to the electrical power system	ObjectReference (IfcRelConnectsElements)	n/a	n/a	NIL
	ConstructionDetails	List of reference to construction detail drawings	LIST [0:?] OF ObjectReference (IfcDocumentReference)	n/a	n/a	empty list
	SpecSection	Reference to a section of the construction specification	ObjectReference (IfcDocumentReference)	n/a	n/a	NIL
	OperationalSpace	Space around fixture required for proper use by occupants	ObjectReference (IfcSpace)	0	N	0
	ManufacturerMaterial	Material selection - from the manufacturer's material options for this fixture type	IfcString	n/a	n/a	empty string
	ManufacturerColor	Color selection - from the manufacturer's color options for this fixture type	IfcString	n/a	n/a	empty string
	ManufacturerFinish	Finish selection - from the manufacturer's finish options for this fixture type	IfcString	n/a	n/a	empty string
	Target objects for this PropertySet					
	IfcPlumbingFixture					
36	**Pset_RailingCommon**					
	ManuafactureInfo	reference to Manufacturer information	ObjectReference (Pset_ManufactureInfo)	n/a	n/a	NIL
	ConstructionDetail	Reference to construction detail drawing	Ref [IfcDocumentReference]	n/a	n/a	NIL
	SpecSection	Reference to a section of the construction specification	ObjectReference (IfcDocumentReference)	n/a	n/a	NIL
	Target objects for this PropertySet					
	IfcRailing					
37	**Pset_RailingHandrail**					
	CommonRailingProperties	Reference to the SharedPropertySet (Pset_RailingCommon). Contains the shared values for this type -- of properties that are stored for all Railing elements.	IfcObjectReference (Pset_RailingCommon)	n/a	n/a	NIL
	HandrailMaterial	Index into the IfcMaterialList defined in the IfcElement supertype	IfcInteger	1	MaterialList length	1
	HandrailHeight	Height to top of handrail - from stair, landing or floor	IfcPositiveLengthMeasure	0.0	see type	0.0
	MaxDistanceFromWall	Distance from the wall to the outside of the handrail. Value of 0.0 means value not set.	IfcPositiveLengthMeasure	0.0	see type	0.0
	Target objects for this PropertySet					
	IfcRailing					
38	**Pset_RailingGuardrail**					
	CommonRailingProperties	Reference to the SharedPropertySet (Pset_RailingCommon). Contains the shared values for this type -- of properties that are stored for all Railing elements.	IfcObjectReference (Pset_RailingCommon)	n/a	n/a	NIL

	Height	Height to the top of the guardrail - from stair, landing or floor	IfcPositiveLengthMeasure	0.0	see type	0.0
	RepeatingElements	reference to definition of repeating rail stiles - defined in a Pset in the ExtensionPsets for this object	Ref [Pset_RepeatingElement]	n/a	n/a	NIL
	MountedHandrail	Reference to any handrail mounted on this guardrail	Ref [IfcRailing]	n/a	n/a	NIL
	Target objects for this PropertySet					
	IfcRailing					
39	**Pset_RailingBalustrade**					
	CommonRailingProperties	Reference to the SharedPropertySet (Pset_RailingCommon). Contains the shared values for this type -- of properties that are stored for all Railing elements.	IfcObjectReference (Pset_RailingCommon)	n/a	n/a	NIL
	BalustradeProperty2	Property not defined yet				
	Target objects for this PropertySet					
	IfcRailing					
40	**Pset_VisualScreenRestroomPartition**					
	CommonVisualScreenElementProperties	Reference to the 'parent' SharedPropertySet (Pset_VisualScreenCommon). Contains the shared values for this type -- of properties that are stored for all VisualScreen elements.	IfcObjectReference (Pset_VisualScreenCommon)	n/a	n/a	NIL
	ManufacturerMaterial	Material selection - from the manufacturer's material options for this fixture type	IfcString	n/a	n/a	empty string
	ManufacturerColor	Color selection - from the manufacturer's color options for this fixture type	IfcString	n/a	n/a	empty string
	ManufacturerFinish	Finish selection - from the manufacturer's finish options for this fixture type	IfcString	n/a	n/a	empty string
	Target objects for this PropertySet					
	IfcVisualScreen					
41	**Pset_VisualScreenRestroomPartitionDoor**					
	CommonVisualScreenElementProperties	Reference to the 'parent' SharedPropertySet (Pset_VisualScreenCommon). Contains the shared values for this type -- of properties that are stored for all VisualScreen elements.	IfcObjectReference (Pset_VisualScreenCommon)	n/a	n/a	NIL
	HingeSide	Indicates the hinged side of the door - when viewed from outside the partition enclosure. 0=left, 1=right.	IfcLogical	0	1	1
	SwingDirection	Indicates whether this door swings into or out of the partition enclosure. 0=swings in, 1=swings out.	IfcLogical	0	1	1
	ManufacturerMaterial	Material selection - from the manufacturer's material options for this fixture type	IfcString	n/a	n/a	empty string
	ManufacturerColor	Color selection - from the manufacturer's color options for this fixture type	IfcString	n/a	n/a	empty string
	ManufacturerFinish	Finish selection - from the manufacturer's finish options for this fixture type	IfcString	n/a	n/a	empty string
	Target objects for this PropertySet					
	IfcVisualScreen					
42	**Pset_VisualScreenCommon**					
	VisualScreenElementHeight	Height of the partition panel. Value of 0.0 means property not set.	IfcPositiveLengthMeasure	0.0	see type	0.0
	VisualScreenElementWidth	Width of the partition panel. Value of 0.0 means property not set.	IfcPositiveLengthMeasure	0.0	see type	0.0
	VisualScreenElementThickness	Thickness of the partition panel. Value of 0.0 means property not set.	IfcPositiveLengthMeasure	0.0	see type	0.0
	AssembledTopOfElementHeight	Height, from finish floor, to the top of this partition panel. Value of 0.0 means property not set.	IfcPositiveLengthMeasure	0.0	see type	0.0
	AssemblyHardware	List of references to mounting/assembly hardware components for this panel only	LIST [0:?] OF ObjectReference (IfcBuiltInAccessories)	n/a	n/a	empty list

	Target objects for this PropertySet						
	IfcVisualScreen						
43	**Pset_VisualScreenAssembly**						
	ManuafactureInfo	reference to Manufacturer information	ObjectReference (Pset_ManufactureInfo)	n/a	n/a	NIL
	ConstructionDetails	List of reference to construction detail drawings	LIST [0:?] OF ObjectReference (IfcDocumentReference)	n/a	n/a	empty list
	SpecSection	Reference to a section of the construction specification	ObjectReference (IfcDocumentReference)	n/a	n/a	NIL
	Target objects for this PropertySet					
	IfcVisualScreen					
44	**Pset_VisualScreenPost**					
	CommonVisualScreenProperties	Reference to the SharedPropertySet (Pset_VisualScreenCommon) - which contains properties that are stored for all types of VisualScreen elements.	IfcSharedPropertySet (Pset_VisualScreenCommon)	n/a	n/a	NIL
	PostProperty2	This property has not yet been defined				
	Target objects for this PropertySet					
	IfcVisualScreen					
45	**Pset_VisualScreenPanel**					
	CommonVisualScreenProperties	Reference to the SharedPropertySet (Pset_VisualScreenCommon) - which contains properties that are stored for all types of VisualScreen elements.	IfcSharedPropertySet (Pset_VisualScreenCommon)	n/a	n/a	NIL
	VisualScreenProperty2	This property has not yet been defined				
	Target objects for this PropertySet					
	IfcVisualScreen					
46	**Pset_VisualScreenDoorOrGate**					
	CommonVisualScreenProperties	Reference to the SharedPropertySet (Pset_VisualScreenCommon) - which contains properties that are stored for all types of VisualScreen elements.	IfcSharedPropertySet (Pset_VisualScreenCommon)	n/a	n/a	NIL
	DoorProperty2	This property has not yet been defined				
	Target objects for this PropertySet					
	IfcVisualScreen					
47	**Pset_Shelf**					
	CommonCounterOrShelfProperties	Reference to the SharedPropertySet (Pset_CounterOrShelfCommon). Contains the shared values for this type -- of properties that are stored for all types of counters and shelves.	IfcObjectReference (Pset_CounterOrShelfCommon)	n/a	n/a	NIL
	Property2	Not yet defined				
	Target objects for this PropertySet					
	IfcCounterOrShelf					
48	**Pset_SpaceElevatorShaft**	Specific type of Technical Space (generic type defined in R1.5)				
	CommonSpaceProperties	Reference to the 'parent' PropertySet (Pset_SpaceCommon). Contains the shared values for this type -- of properties that are stored for all Spaces.	IfcObjectReference (Pset_SpaceCommon)	n/a	n/a	NIL
	BldgStoriesServiced	List of references to all the IfcBuildinStorey objects that are serviced by this elevator shaft	LIST [0:?] OF Ref [IfcBuildingStorey]	n/a	n/a	empty list
	Target objects for this PropertySet					

				IfcSpace					
49		**Pset_SpaceStairShaft**							
				CommonSpaceProperties	Reference to the 'parent' PropertySet (Pset_SpaceCommon). Contains the shared values for this type -- of properties that are stored for all Spaces.	IfcObjectReference (Pset_SpaceCommon)	n/a	n/a	NIL
				BldgStoriesServiced	List of references to all the IfcBuildinStorey objects that are serviced by this elevator shaft	LIST [0:?] OF Ref [IfcBuildingStorey]	n/a	n/a	empty list
			Target objects for this PropertySet						
				IfcSpace					
50		**Pset_WallParapet**							
				CommonWallProperties	Reference to the 'parent' PropertySet (Pset_WallType). Contains the shared values for this type -- of properties that are stored for all walls.	IfcObjectReference (Pset_WallType)	n/a	n/a	NIL
				RepeatingElements	Reference to a Pset (Pset_RepeatingElement) describing any repeating elements for this wall (NIL pointer if not)	IfcObjectReference (Pset_RepeatingElement)	n/a	n/a	NIL
				CapMaterial	Index into the Material list defined at the IfcElement level - points to material from which cap is made.	IfcInteger	see type	see type	1
				CapManuafacturerInfo	Reference to a SharedPropertySet - Pset_ManufactureInformation, which defines information about the manufacture of the parapet cap.	IfcObjectReference (Pset_ManufactureInformation)	n/a	n/a	NIL
				ConstructionDetails	Document references to detail drawings	LIST [0:?] OF IfcDocumentReference	n/a	n/a	empty list
				ParapetFunctionTypeEnum	Reference to a Pset enumerating possible parapet functional roles. Note: this will be implemented as a shared PropertySet (Pset_ParapetFunctionTypeEnum) of IfcString (values enumerated at right).	IfcObjectReference (Pset_ParapetFunctionTypeEnum - ENUMERATION OF (window washing rigging support, handrail, screen, fire block))	n/a	n/a	NIL
				ParapetFunctionTypeIndex	List of Integer indicies into the enumeration defined by Pset_ParapetFunctionTypeEnum.	LIST [1:?] OF IfcInteger	1	4	empty list
				SpecSection	Reference to relevant section of the construction specifications	ObjectReferenc (IfcDocumentReference)	n/a	n/a	empty list
			Target objects for this PropertySet						
				IfcWall					
51		**Pset_WindowSkylight**							
				CommonWindowProperties	Reference to the 'parent' PropertySet (Pset_WindowType). Contains the shared values for this type -- of properties that are stored for all windows.	IfcSharedPropertySet (Pset_WindowType)	n/a	n/a	NIL
				WindowPanelList	Reference to one or more window panels (defined left to right or bottom to top), as viewed from the finished (exterior) face (see diagram in specificaitons). NOTES: 1) this will be implemented as a shared Pset_WindowPanelList - which contains a list of s	LIST [1:?] OF IfcObjectReference (Pset_WindowPanel)	n/a	n/a	NIL
				Operable	Is this Skylight operable?	IfcBoolean	FALSE	TRUE	FALSE
			Target objects for this PropertySet						
				IfcWindow					

6.2. [AR-2] Compartmentation of Buildings

6.2.1. Object Types

The following table is pasted from the speadsheet template "R2_ObjectDefs_d4.xls", sheet "Class Definitions"

		Object Type Name	Interface name		OPT INV DER					
Subtype of				Attribute / Relation name		Data Type/Related Object Type	Min	Max	Default	Definition
IfcCharacteristic	1	**IfcSpaceOccupancy**								IfcSharedSpatialElements schema
			I_SpaceOccupancy							
				OccupancyNumber	OPT	IfcOccupancyNumber	see type	see type	see type	
				Owner	OPT	IfcOccupant	see type	see type	see type	
				Rental	OPT	IfcOccupant	see type	see type	see type	
				Lease	OPT	IfcOccupant	see type	see type	see type	
IfcCharacteristic	2	**IfcOccupant**								IfcSharedSpatialElements schema
			I_Occupant							
				GenericType		IfcOccupantTypeEnum	Owner	Lessee	Tenant	
				OccupantName	OPT	IfcActorSelect				
IfcCharacteristic	3	**IfcOccupancyNumber**								IfcSharedSpatialElements schema
			I_OccupancyNumber							
				GenericType		IfcOccupancyNumber TypeEnum	Number	Number	Number	
				ActualOccupancyNumber	OPT	INTEGER	0	see type	1	
				DesignIntentOccupancyNu mber	OPT	INTEGER	0	see type	1	
				CumulativeOccupancyNum ber	OPT	INTEGER	0	see type	1	
				OccupancyRate	OPT	IfcPersonPerAreaMea sure	0	see type	1	
IfcSpace	4	**IfcFireCompartment**								IfcSharedSpatialElements schema
			I_FireCompartment							
				calcHeightAboveGrade	OPT	IfcLengthMeasure	see type	see type	1.0	
				MainFireUse	OPT	IfcClassification	see type	see type	see type	Main fire use for the space which is assigned from the Fire Use Classification.
				AncillaryFireUse	OPT	IfcClassification	see type	see type	see type	Ancillary fire use for the space which is assigned from the Fire Use Classification.
				FireRiskFactor	OPT	INTEGER	see type	see type	see type	Fire Risk factor assigned to the space
				NaturalVentilation	OPT	BOOLEAN	see type	see type	see type	Indication whether the space is ventilated natural (true) or mechanical (false).
				SprinklerProtection	OPT	BOOLEAN	see type	see type	see type	Indication whether the space is sprinkler protected (true) or not (false).

6.2.2. Type Definitions

None were defined.

6.2.3. Property Sets

None were defined.

6.3. [BS-1] HVAC System Design

6.3.1. Object Types

The following table is pasted from the speadsheet template "R2_ObjectDefs_d4.xls", sheet "Class Definitions"

Class Name		{{ "Ref" = relationship }}				
Interface name		*Data Type*	*Min*	*Max*	*Default*	*Units*
IfcPathwayElement	*This class connects together the parts of a networked system.*					
InheritsFrom ---------->	*IfcBuildingElement*					
I_PathwayElement						
PathwayElementType	*Named type of PathwayElement. References a PathwayElement TypeDef which links to attributes shared by all instances of this type.*	*Ref [IfcTypeDefinition]*	*n/a*	*n/a*	*NIL*	*n/a*
InletPointConnections	*Specifies which IfcPointConnectors are inlets. All remaining IfcPointConnectors are therefore outlets.*	*List [0:N] Ref [IfcPointConnector]*	*n/a*	*n/a*	*NIL*	*n/a*
MaterialLayerSet	*Material layer set the pathway element is constructed with. We use a material layer set to allow multiple materials to be used to construct a pathway element.*	*Ref [IfcMaterialLayerSet]*	*n/a*	*n/a*	*NIL*	*n/a*
IsMountedOn	*IfcObject that the device is mounted upon or attached to, such as a wall or structural support. This relationship allows the PathwayElement to appropriately move if the object it is 'mounted' upon is moved, while maintaining its system interconnectivity.*	*Ref[IfcObject]*	*n/a*	*n/a*	*NIL*	*n/a*
Description	*A user-defined string description of the pathway element*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
IfcDamper	*This class is used to control or reduce air flow in a duct system.*					
InheritsFrom ---------->	*IfcPathwayElement*					
I_Damper						
DamperType	*Named type of Damper References a Damper TypeDef which links to attributes shared by all instances of this type.*	*Ref [IfcTypeDefinition]*	*n/a*	*n/a*	*NIL*	*n/a*
Inlet Connection	*Inlet Connection references a Size AttDef which contains the shape and size of the connection (e.g., Att_RoundDuctConnection)*	*Ref [IfcAttDef]*	*n/a*	*n/a*	*NIL*	*n/a*
Outlet Connection	*Outlet Connection references a Size AttDef which contains the shape and size of the connection (e.g., Att_RoundDuctConnection)*	*Ref [IfcAttDef]*	*n/a*	*n/a*	*NIL*	*n/a*
Frame Depth	*The length (or depth) of the damper frame*	*IfcLengthMeasure*	*see type*	*see type*	*0.0*	*n/a*
SizingMethod	*Enumeration that identifies whether the damper is sized nominally or with exact measurements*	*Enum [Nominal, Exact]*	*n/a*	*n/a*	*Nominal*	*n/a*
Manufacturer	*The manufacturer of the damper assembly*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
Model	*The manufacturer's model number of the damper assembly*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
WorkingPressure	*The actual working pressure of the damper assembly*	*IfcPressureMeasure*	*see type*	*see type*	*0.0*	*n/a*

DesignPressureRating	*The design pressure rating for the damper assembly*	*IfcPressureMeasure*	*see type*	*see type*	*0.0*	*n/a*
PressureLoss	*The pressure loss across the damper assembly*	*IfcPressureMeasure*	*see type*	*see type*	*0.0*	*n/a*
CloseOffRating	*Close off rating*	*IfcPressureMeasure*	*see type*	*see type*	*0.0*	*n/a*
LeakageAirFlowrate	*Leakage air flow rate*	*IfcVolumetricFlowrateMeasure*	*see type*	*see type*	*0.0*	*n/a*
IfcControlElement	*This class is used to identify control components that are typically a part of any HVAC duct or piping system.*					
InheritsFrom ---------->	*IfcPathwayElement*					
I_ControlElement						
ControlElementType	*Named type of ControlElement references a ControlElement TypeDef which links to attributes shared by all instances of this type.*	*Ref [IfcTypeDefinition]*	*n/a*	*n/a*	*NIL*	*n/a*
BACnetObjectType	*The BACnet object type of the device*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
PointID	*The Point Identification assigned to this control element*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
IfcActuator	*This class is used to identify actuators.*					
InheritsFrom ---------->	*IfcControlElement*					
I_Actuator						
ActuatorType	*Named type of Actuator; references an Actuator TypeDef which links to attributes shared by all instances of this type.*	*Ref [IfcTypeDefinition]*	*n/a*	*n/a*	*NIL*	*n/a*
ActuatorFunction	*Named type of Actuator references an Actuator TypeDef which links to attributes which vary by type.*	*Ref [IfcTypeDefinition]*	*n/a*	*n/a*	*NIL*	*n/a*
Operation	*Enumeration that identifies the type of motion the actuator provides*	*Enum [Modulating, TwoPosition]*	*n/a*	*n/a*	*Modulating*	*n/a*
FailPosition	*Enumeration that identifies the behavior of the actuator in case of power failure*	*Enum [FailOpen, FailClosed, None]*	*n/a*	*n/a*	*FailOpen*	*n/a*

6.3.2. Type Definitions

The following table is pasted from the speadsheet template "R2_ObjectDefs_d4.xls", sheet "Type Definitions"

	#			Description
		TypeDef Name		
		Class being Typed		
		Generic Type		
Defined		*Specific Type / project defining type*		
In				*Property Sets*
	PreDefined Type Definitions for IFC Release 2			
BS-1		**PathwayElementType**		*Allows definition of defined types of pathway elements*
		IfcPathwayElement		
		AirTerminal		
		< Any value >		
		BS-1	*shared =*	*Pset_AirTerminal*
			occurrence =	*<none defined>*
		Damper		
		< Any value >		
		BS-1	*shared =*	*Pset_Damper*
			occurrence =	*<none defined>*
		TerminalBox		
		< Any value >		
		BS-1	*shared =*	*Pset_TerminalBox*

								occurrence =	*<none defined>*
					DuctFitting				
						< Any value >			
						BS-1		*shared =*	*Pset_DuctFitting*
								occurrence =	*<none defined>*
					DuctSegment				
						< Any value >			
						BS-1		*shared =*	*Pset_DuctSegment*
								occurrence =	*<none defined>*
					Valve				
						< Any value >			
						BS-1		*shared =*	*Pset_Valve*
								occurrence =	*<none defined>*
					PipeFitting				
						< Any value >			
						BS-1		*shared =*	*Pset_PipeFitting*
								occurrence =	*<none defined>*
					PipeSegment				
						< Any value >			
						BS-1		*shared =*	*Pset_PipeSegment*
								occurrence =	*<none defined>*
BS-1			**Damper**						*Allows definition of defined types of dampers*
				IfcDamper					
					FireDamper				
						< Any value >			
						BS-1		*shared =*	*Pset_FireDamper*
								occurrence =	*<none defined>*
					SmokeDamper				
						< Any value >			
						BS-1		*shared =*	*Pset_SmokeDamper*
								occurrence =	*<none defined>*
					FireSmokeDamper				
						< Any value >			
						BS-1		*shared =*	*Pset_FireSmokeDamper*
								occurrence =	*<none defined>*
					BackdraftDamper				
						< Any value >			
						BS-1		*shared =*	*Pset_BackdraftDamper*
								occurrence =	*<none defined>*
					ControlDamper				
						< Any value >			
						BS-1		*shared =*	*Pset_ControlDamper*
						BS-1		*occurrence =*	*<none defined>*
					Louver				
						< Any value >			
						BS-1		*shared =*	*Pset_Louver*
						BS-1		*occurrence =*	*<none defined>*
BS-1			**ControlElement**						*Allows definition of defined types of control elements*
				IfcControlElement					

					Controller			
						< Any value >		
						BS-1	*shared =*	*Pset_Controller*
							occurrence =	*<none defined>*
					Sensor			
						< Any value >		
						BS-1	*shared =*	*Pset_Sensor*
							occurrence =	*<none defined>*
BS-1			**Actuator**					*Allows definition of defined types of actuators*
				IfcControlElement				
					LinearActuator			
						< Any value >		
						BS-1	*shared =*	*Pset_LinearActuator*
						BS-1	*occurrence =*	*Pset_ElectricActuator, Pset_PneumaticActuator, Pset_HydraulicActuator, Pset_HandOperatedActuator*
					RotationalActuator			
						< Any value >		
						BS-1	*shared =*	*Pset_RotationalActuator*
						BS-1	*occurrence =*	*Pset_ElectricActuator, Pset_PneumaticActuator, Pset_HydraulicActuator, Pset_HandOperatedActuator*

6.3.3. Property Sets

The following table is pasted from the speadsheet template "R3_ObjectDefs_d4.xls", sheet "PropertySet Definitions."

Property Set Name / Property name	Definition	Data Type	Min	Max	Default	Units
Property Set Name						
Property name	***Definition***	***Data Type***	***Min***	***Max***	***Default***	***Units***
PreDefined Property Sets for IFC Release 2						
Note: These tables define property sets for 3 purposes						
1. Shared property sets associated with a type (see TypeDefinition above)						
2. Variable property sets associated with a type (those which vary for each occurrence)						
3. Domain extension model extensions to classes in the IFC Core						
This table is divided into 3 parts, according to the purpose of the property set						
Shared Property Sets defining Type						
BS-1 Model						
PathwayElement	*PointConnections, combined with the information in a connection type property set (i.e., Pset_RectangularDuctConnection, Pset_RoundDuctConnection) attached to the referenced IfcPointConnector, provide the required information for the type, size and location of physical connections.*					
Pset_AirTerminal						
Purpose	*This property set will be used by an IfcPathwayElement object for defining Air Terminals*					
AirTerminalType	*Enumeration defining the type of Air Terminal*	*Enum [Supply, Return, Exhaust, Other]*	*n/a*	*n/a*	*Supply*	*n/a*
Flowrate	*Maximum air flowrate for the terminal device*	*IfcVolumetricFlowrateMeasure*	*see type*	*see type*	*0.0*	*n/a*
PressureLoss	*Pressure loss through the terminal device*	*IfcPressureMeasure*	*see type*	*see type*	*0.0*	*n/a*
Throw	*The distance the air terminal throws the air (optional)*	*IfcLengthMeasure*	*see type*	*see type*	*0.0*	*n/a*
SoundLevel	*Design sound power level*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
ADPI	*Air diffusion performance index*	*REAL*	*0.0*	*0.0*	*0.0*	*n/a*
InletConnection	*Reference to the size and shape of the inlet connection (e.g., Pset_RoundDuctConnection)*	*Ref [IfcAttDef]*	*see type*	*see type*	*NIL*	*n/a*
OutletConnection	*Reference to the size and shape of the outlet connection (e.g., Pset_RoundDuctConnection)*	*Ref [IfcAttDef]*	*see type*	*see type*	*NIL*	*n/a*

FinishType	*Enumeration that identifies the type of finish for the air terminal*	*Enum [Anodized, Paint, None]*	*n/a*	*n/a*	*None*	*n/a*
FinishColor	*The finish color for the air terminal*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
MountingFrame	*Frame for plaster, drywall, lay-in grid, etc.*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
AdjustableCore	*Permits adjustment of throw*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
CoreSetHorizontal	*Degree of blade set from the centerline*	*IfcPlaneAngleMeasure*	*see type*	*see type*	*0.0*	*n/a*
CoreSetVertical	*Degree of blade set from the centerline*	*IfcPlaneAngleMeasure*	*see type*	*see type*	*0.0*	*n/a*
IntegralDamper	*Reference to a damper object that is integral to the terminal device*	*Ref [IfcDamper]*	*see type*	*see type*	*NIL*	*n/a*
IntegralControl	*Self powered temperature control*	*BOOL*	*FALSE*	*TRUE*	*FALSE*	*n/a*
Pset_TerminalBox						
Purpose	*This property set will be used by an IfcPathwayElement object to define Terminal Boxes*					
TerminalBoxType	*Enumeration that identifies the type of terminal box: VAV, CV, VVRH, etc.*	*Enum [VAV, CV, VAVReheat, CVReheat, FanPowered, VAVDualDuct, CVDualDuct]*	*n/a*	*n/a*	*VAV*	*n/a*
DesignFlowrate	*Maximum air flowrate for the terminal box*	*IfcVolumetricFlowrateMeasure*	*see type*	*see type*	*0.0*	*n/a*
MinimumFlowrate	*Minimum air flowrate for the terminal box*	*IfcVolumetricFlowrateMeasure*	*see type*	*see type*	*0.0*	*n/a*
PressureLoss	*Pressure loss through the terminal box*	*IfcPressureMeasure*	*see type*	*see type*	*0.0*	*n/a*
SoundLevel	*Design sound power level*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
Pset_DuctFitting						
Purpose	*This property set will be used by an IfcPathwayElement object to define duct fittings. This property set is used in conjunction with an Pset_DuctDesignCriteria which defines common duct design parameters.*					
PrimaryType	*Enumeration that identifies the primary type of fitting (i.e., elbow, transition, junction, etc.)*	*Enum [Entry, Exit, Elbow, Transition, Junction, Obstruction, Hood, Other]*	*n/a*	*n/a*	*Elbow*	*n/a*
SubType	*Subtype of fitting (i.e., 5-gore, pleated, stamped, etc.)*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
EnteringPressure	*Actual pressure required for balancing and maintenance*	*IfcPressureMeasure*	*see type*	*see type*	*0.0*	*n/a*
Angle	*Angle of turn for elbows, transitions, etc.*	*IfcPlaneAngleMeasure*	*see type*	*see type*	*0.0*	*n/a*
Pset_DuctSegment						
Purpose	*This property set will be used by an IfcPathwayElement object to define duct segments. This property set is used in conjunction with an Pset_DuctDesignCriteria which defines common duct design parameters*					
Flowrate	*Flowrate through the duct*	*IfcVolumetricFlowrateMeasure*	*see type*	*see type*	*0.0*	*n/a*
EnteringPressure	*Actual pressure required for balancing and maintenance*	*IfcPressureMeasure*	*see type*	*see type*	*0.0*	*n/a*
SupportMethod	*Reference to a duct hanger or other structural support from roof, floor, etc.*	*Ref [IfcObject]*	*see type*	*see type*	*NIL*	*n/a*
FinishedLength	*The finished length of the duct segment*	*IfcLengthMeasure*	*see type*	*see type*	*0.0*	*n/a*
LongitudinalSeam	*The type of seam to be used along the longitudinal axis of the duct segment*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
Reinforcement	*The type of reinforcement used for the duct segment*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
ReinforcementSpacing	*The spacing between reinforcing elements*	*IfcLengthMeasure*	*see type*	*see type*	*0.0*	*n/a*
Pset_Valve						
Purpose	*This property set will be used by an IfcPathwayElement object to define valves*					
WorkingPressure	*Working pressure*	*IfcPressureMeasure*	*see type*	*see type*	*0.0*	*n/a*
PressureDrop	*Pressure drop*	*IfcPressureMeasure*	*see type*	*see type*	*0.0*	*n/a*
CloseOffRating	*Close off rating*	*IfcPressureMeasure*	*see type*	*see type*	*0.0*	*n/a*
ValveCv	*Cv value for the valve*	*REAL*	*0.0*	*0.0*	*0.0*	*n/a*
Pset_PipeFitting						
Purpose	*This property set will be used by an IfcPathwayElement object to define pipe fittings. This property set is used in conjunction with an Pset_PipeDesignCriteria which defines common pipe design parameters*					
PrimaryType	*Enumeration that identifies the primary type of fitting (i.e., elbow, transition, junction, etc.)*	*Enum [Entry, Exit, Elbow, Transition, Junction, Obstruction, Other]*	*n/a*	*n/a*	*Elbow*	*n/a*
SubType	*Subtype of fitting (i.e., long-radius, short-radius,*	*IfcString*	*see type*	*see type*	*empty*	*n/a*

					etc.)				*string*	
				EnteringPressure	*Actual pressure required for balancing and maintenance*	*IfcPressureMeasure*	*see type*	*see type*	*0.0*	*n/a*
				Angle	*Angle of turn for elbows, transitions, etc.*	*IfcPlaneAngleMeasure*	*see type*	*see type*	*0.0*	*n/a*
		Pset_PipeSegment								
			Purpose		*This property set will be used by an IfcPathwayElement object to define pipe segments. This property set is used in conjunction with an Pset_PipeDesignCriteria which defines common pipe design parameters*					
				Flowrate	*Flowrate through the pipe*	*IfcVolumetricFlowrateMeasure*	*see type*	*see type*	*0.0*	*n/a*
				EnteringPressure	*Actual pressure required for balancing and maintenance*	*IfcPressureMeasure*	*see type*	*see type*	*0.0*	*n/a*
				SupportMethod	*Reference to a pipe hanger or other structural support from roof, floor, etc.*	*Ref [IfcObject]*	*see type*	*see type*	*NIL*	*n/a*
				FinishedLength	*The finished length of the pipe segment*	*IfcLengthMeasure*	*see type*	*see type*	*0.0*	*n/a*
		Pset_FireDamper								
			Purpose		*This property set will be used by an IfcDamper object to define the characteristics of a fire damper*					
				ClosureRating	*Enumeration that identifies the closure rating for the damper*	*Enum [Dynamic, Static]*	*n/a*	*n/a*	*Dynamic*	*n/a*
				FireResistanceRating	*Enumeration that identifies the fire resistance rating of the damper*	*Enum [1-1/2Hour, 3Hour]*	*n/a*	*n/a*	*1-1/2Hour*	*n/a*
				Mounting	*Enumeration that identifies how the damper is mounted in the building*	*Enum [Horizontal, Vertical]*	*n/a*	*n/a*	*Vertical*	*n/a*
				FusibleLinkTemperature	*The temperature that the fusible link melts*	*IfcThermodynamicTemperatureMeasure*	*see type*	*see type*	*0.0*	*n/a*
				SleeveLength	*The length of the damper sleeve*	*IfcLengthMeasure*	*see type*	*see type*	*0.0*	*n/a*
				SleeveThickness	*The thickness of the damper sleeve*	*IfcLengthMeasure*	*see type*	*see type*	*0.0*	*n/a*
				DamperLocationInSleeve	*The location within the sleeve where the damper is mounted (e.g., Center)*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
		Pset_SmokeDamper								
			Purpose		*This property set will be used by an IfcDamper object to define the characteristics of a smoke damper*					
				FrameThickness	*The thickness of the damper frame*	*IfcLengthMeasure*	*see type*	*see type*	*0.0*	*n/a*
				BladeType	*The type of blade used in the damper (e.g., Triple Vee, Fabricated Airfoil, Extruded Airfoil, etc.)*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
				Mounting	*Enumeration that identifies how the damper is mounted in the building*	*Enum [Horizontal, Vertical]*	*n/a*	*n/a*	*Vertical*	*n/a*
				ControlType	*The type of control used to operate the damper (e.g., Open/Closed Indicator, Resetable Temperature Sensor, Temperature Override, etc.)*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
				SleeveLength	*The length of the damper sleeve*	*IfcLengthMeasure*	*see type*	*see type*	*0.0*	*n/a*
				SleeveThickness	*The thickness of the damper sleeve*	*IfcLengthMeasure*	*see type*	*see type*	*0.0*	*n/a*
				DamperLocationInSleeve	*The location within the sleeve where the damper is mounted (e.g., Center)*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
				Actuator	*Actuator references an Pset_Actuator AttDef which contains the actuator information, if an actuator is part of the damper assembly*	*Ref [IfcAttDef]*	*n/a*	*n/a*	*NIL*	*n/a*
		Pset_FireSmokeDamper								
			Purpose		*This property set will be used by an IfcDamper object to define the characteristics of a combination smoke and fire damper*					
				FrameThickness	*The thickness of the damper frame*	*IfcLengthMeasure*	*see type*	*see type*	*0.0*	*n/a*
				FireResistanceRating	*Enumeration that identifies the fire resistance rating of the damper*	*Enum [1-1/2Hour, 3Hour]*	*n/a*	*n/a*	*1-1/2Hour*	*n/a*
				BladeType	*The type of blade used in the damper (e.g., Triple Vee, Fabricated Airfoil, Extruded Airfoil, etc.)*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
				Mounting	*Enumeration that identifies how the damper is mounted in the building*	*Enum [Horizontal, Vertical]*	*n/a*	*n/a*	*Vertical*	*n/a*
				FusibleLinkTemperature	*The temperature that the fusible link melts*	*IfcThermodynamicTemperatureMeasure*	*see type*	*see type*	*0.0*	*n/a*
				ControlType	*The type of control used to operate the damper (e.g., Open/Closed Indicator, Resetable Temperature Sensor, Temperature Override, etc.)*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
				SleeveLength	*The length of the damper sleeve*	*IfcLengthMeasure*	*see type*	*see type*	*0.0*	*n/a*
				SleeveThickness	*The thickness of the damper sleeve*	*IfcLengthMeasure*	*see type*	*see type*	*0.0*	*n/a*
				DamperLocationInSleeve	*The location within the sleeve where the damper is mounted (e.g., Center)*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
				Actuator	*Actuator references an Pset_Actuator AttDef which contains the actuator information, if an actuator is part of the damper assembly*	*Ref [IfcAttDef]*	*n/a*	*n/a*	*NIL*	*n/a*
		Pset_BackdraftDamper								
			Purpose		*This property set will be used by an IfcDamper object to define the characteristics of a backdraft damper*					
				FrameType	*The type of frame used by the damper (e.g., Standard, Single Flange, Single Reversed Flange,*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*

	Double Flange, etc.)					
Actuator	*Actuator references an Pset_Actuator AttDef which contains the actuator information, if an actuator is part of the damper assembly*	*Ref [IfcAttDef]*	*n/a*	*n/a*	*NIL*	*n/a*
Pset_ControlDamper						
Purpose	*This property set will be used by an IfcDamper object to define the characteristics of a control damper*					
DesignAirVelocity	*The design air velocity for the damper assembly*	*IfcLinearVelocityMeasure*	*see type*	*see type*	*0.0*	*n/a*
BladeAction	*Enumeration that identifies the blade closing action for the damper*	*Enum [Parallel, Opposed]*	*n/a*	*n/a*	*Parallel*	*n/a*
BladeType	*The type of blade used in the damper (e.g., Triple Vee, Fabricated Airfoil, Extruded Airfoil, etc.)*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
BladeMaterial	*The primary material used to construct the damper blade*	*Ref [IfcMaterial]*	*n/a*	*n/a*	*NIL*	*n/a*
BladeThickness	*The thickness of the damper blade*	*IfcLengthMeasure*	*see type*	*see type*	*0.0*	*n/a*
FrameType	*The type of frame used by the damper (e.g., Standard, Single Flange, Single Reversed Flange, Double Flange, etc.)*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
FrameMaterial	*The primary material used to construct the damper frame*	*Ref [IfcMaterial]*	*n/a*	*n/a*	*NIL*	*n/a*
FrameThickness	*The thickness of the damper frame*	*IfcLengthMeasure l*	*see type*	*see type*	*0.0*	*n/a*
Actuator	*Actuator references an Pset_Actuator AttDef which contains the actuator information, if an actuator is part of the damper assembly*	*Ref [IfcAttDef]*	*n/a*	*n/a*	*NIL*	*n/a*
Pset_Louver						
Purpose	*This property set will be used by an IfcDamper object to define the characteristics of a louver*					
FrameType	*The type of frame used by the louver (e.g., Standard, Drainable, etc.)*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
FrameThickness	*The thickness of the louver frame*	*IfcLengthMeasure*	*see type*	*see type*	*0.0*	*n/a*
BladeType	*The type of blade used in the louver (e.g., "J", "K", Cheveron, Sightproof, Drainable, etc.)*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
BladeThickness	*The thickness of the louver blade*	*IfcLengthMeasure*	*see type*	*see type*	*0.0*	*n/a*
ScreenType	*The type of screen used in the louver (e.g., Birdscreen, Insect Screen, etc.)*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
Actuator	*Actuator references an Pset_Actuator AttDef which contains the actuator information, if an actuator is part of the louver assembly*	*Ref [IfcAttDef]*	*n/a*	*n/a*	*NIL*	*n/a*
Pset_LinearActuator						
Purpose	*This property set will be used by an IfcControlElement object to define the characteristics of a linear actuator*					
FailDirection	*Enumeration that identifies the behavior of the actuator in case of power failure*	*Enum [FailIn, FailOut]*	*n/a*	*n/a*	*FailIn*	*n/a*
Force	*Indicates the maximum close-off force for the actuator*	*IfcForceMeasure*	*see type*	*see type*	*0.0*	*n/a*
Stroke	*Indicates the maximum distance the actuator must traverse*	*IfcLengthMeasure*	*see type*	*see type*	*0.0*	*n/a*
Pset_RotationalActuator						
Purpose	*This property set will be used by an IfcControlElement object to define the characteristics of a rotational actuator*					
FailDirection	*Enumeration that identifies the behavior of the actuator in case of power failure*	*Enum [FailClockwise, FailCounterClockwise]*	*n/a*	*n/a*	*FailClockwise*	*n/a*
Torque	*Indicates the maximum close-off torque for the actuator*	*IfcTorqueMeasure*	*see type*	*see type*	*0.0*	*n/a*
Range	*Indicates the maximum rotation the actuator must traverse*	*IfcPlaneAngleMeasure*	*see type*	*see type*	*0.0*	*n/a*
Pset_Sensor						
Purpose	*This property set will be used by an IfcControlElement object to define the characteristics of a sensor*					
SensorType	*Enumeration that identifies the type of sensor*	*Enum [Flow, Pressure, Temperature, Gas, Concentration, Volts, Amps, Density, Viscosity, Energy, Humidity, Other]*	*n/a*	*n/a*	*Flow*	*n/a*
SensorDescription	*Further elaboration on the type of sensor*	*IfcString*	*n/a*	*n/a*	*empty string*	*n/a*
SensedMedium	*The medium interacting with the sensor*	*Ref[Pset_Fluid]*	*n/a*	*n/a*	*NIL*	*n/a*
SensorRange	*The range of the sensor*	*REAL*	*0.0*	*0.0*	*0.0*	*n/a*
SensorAccuracy	*The accuracy of the sensor*	*REAL*	*0.0*	*0.0*	*0.0*	*n/a*
Pset_Controller						
Purpose	*This property set will be used by an IfcControlElement object to define the characteristics of a controller*					

InputObjects	*Objects having an input to the controller*	*List [0:N] Ref [IfcObject]*	*n/a*	*n/a*	*NIL*	*n/a*
OutputObjects	*Objects that have an output from the controller*	*List [0:N] Ref [IfcObject]*	*n/a*	*n/a*	*NIL*	*n/a*
Pset_ControlElementAnalogInput						
Purpose	*Analog input for a control element*					
Units	*BACnetEngineeringUnits definition*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
HighLimit	*The high limit value (optional)*	*REAL*	*0.0*	*0.0*	*0.0*	*n/a*
LowLimit	*The low limit value (optional)*	*REAL*	*0.0*	*0.0*	*0.0*	*n/a*
Deadband	*The deadband value (optional)*	*REAL*	*0.0*	*0.0*	*0.0*	*n/a*
LimitEnable	*BACnetLimitEnable definition (optional)*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
EventEnable	*BACnetEventTransitionBits definition (optional)*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
NotifyType	*BACnetNotifyType definition (optional)*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
Pset_ControlElementAnalogOutput						
Purpose	*Analog output for a control element*					
Units	*BACnetEngineeringUnits definition*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
HighLimit	*The high limit value (optional)*	*REAL*	*0.0*	*0.0*	*0.0*	*n/a*
LowLimit	*The low limit value (optional)*	*REAL*	*0.0*	*0.0*	*0.0*	*n/a*
Deadband	*The deadband value (optional)*	*REAL*	*0.0*	*0.0*	*0.0*	*n/a*
LimitEnable	*BACnetLimitEnable definition (optional)*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
EventEnable	*BACnetEventTransitionBits definition (optional)*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
NotifyType	*BACnetNotifyType definition (optional)*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
Pset_ControlElementBinaryInput						
Purpose	*Binary input for a control element*					
Polarity	*BACnetPolarity definition*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
InactiveText	*Inactive Text (optional)*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
ActiveText	*Active Text (optional)*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
AlarmValue	*BACnetBinaryPV definition (optional)*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
EventEnable	*BACnetEventTransitionBits definition (optional)*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
AckedTransitions	*BACnetEventTransitionBits definition (optional)*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
Pset_ControlElementBinaryOutput						
Purpose	*Binary output for a control element*					
Polarity	*BACnetPolarity definition*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
InactiveText	*Inactive Text (optional)*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
ActiveText	*Active Text (optional)*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
MinimumOffTime	*Minimum Off Time (optional)*	*IfcLocalTime*	*see type*	*see type*	*0:0*	*n/a*
MinimumOnTime	*Minimum On Time (optional)*	*IfcLocalTime*	*see type*	*see type*	*0:0*	*n/a*
FeedbackValue	*BACnetBinaryPV definition (optional)*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
EventEnable	*BACnetEventTransitionBits definition (optional)*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
AckedTransitions	*BACnetEventTransitionBits definition (optional)*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
Pset_ControlElementMultiStateInput						
Purpose	*Multi-state input for a control element*					
NumberOfStates		*INT*	*0*	*32726*	*0*	*n/a*
StateText		*List [0:?] IfcString*	*see type*	*see type*	*empty string*	*n/a*
AlarmValues	*(optional)*	*List [0:?] REAL*	*0.0*	*0.0*	*0.0*	*n/a*
EventEnable	*BACnetEventTransitionBits definition (optional)*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
NotifyType	*BACnetNotifyType definition (optional)*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
Pset_ControlElementMultiStateOutput						

Purpose	*Multi-state output for a control element*					
NumberOfStates		*INT*	*0*	*32726*	*0*	*n/a*
StateText		*List [0:?] IfcString*	*see type*	*see type*	*empty string*	*n/a*
EventEnable	*BACnetEventTransitionBits definition (optional)*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
NotifyType	*BACnetNotifyType definition (optional)*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
Pset_ControlElementEventEnrollment						
Purpose	*The events which a control element participates*					
EventType	*BACnetEventType definition*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
NotifyType	*BACnetNotifyType definition*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
EventParameters	*BACnetEventParameter definition*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
ObjectPropertyReference	*BACnetObjectPropertyReference definition*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
EventEnable	*BACnetEventTransitionBits definition*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
Recipient	*BACnetRecipient definition (optional)*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
ProcessIdentifier	*BACnetEventTransitionBits definition (optional)*	*REAL*	*0.0*	*0.0*	*0.0*	*n/a*
IssueConfirmedNotifications	*(optional)*	*BOOL*	*FALSE*	*TRUE*	*FALSE*	*n/a*
Pset_ControlElementLoop						
Purpose	*Loop definition for a control element*					
NumberOfStates		*INT*	*0*	*32726*	*0*	*n/a*
OutputUnits	*BACnetEngineeringUnits definition*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
ManipulatedVariableReference	*BACnetObjectPropertyReference definition*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
ControlledVariableReference	*BACnetObjectPropertyReference definition*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
ControlledVariableUnits	*BACnetEngineeringUnits definition*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
SetpointReference	*BACnetSetpointReference definition*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
Action	*BACnetAction definition*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
PriorityForWriting		*List [0:?] REAL*	*0.0*	*0.0*	*0.0*	*n/a*
EventEnable	*BACnetEventTransitionBits definition (optional)*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
NotifyType	*BACnetNotifyType definition (optional)*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*
Type driven properties that vary for each occurrence						
BS-1 Model						
Control Elements						
Pset_ElectricActuator						
Purpose	*This property set will be used by an IfcControlElement object to define the characteristics of an electric actuator*					
ManualOverride	*Identifies whether hand-operated operation is provided as an override*	*BOOL*	*FALSE*	*TRUE*	*FALSE*	*n/a*
InputPower	*Maximum input power requirement*	*IfcPowerMeasure*	*see type*	*see type*	*0.0*	*n/a*
Pset_PneumaticActuator						
Purpose	*This property set will be used by an IfcControlElement object to define the characteristics of a pneumatic actuator*					
ManualOverride	*Identifies whether hand-operated operation is provided as an override*	*BOOL*	*FALSE*	*TRUE*	*FALSE*	*n/a*
InputPressure	*Maximum input control air pressure requirement*	*IfcPressureMeasure*	*see type*	*see type*	*0.0*	*n/a*
InputFlowrate	*Maximum input control air flowrate requirement*	*IfcVolumetricFlowrateMeasure*	*see type*	*see type*	*0.0*	*n/a*
Pset_HydraulicActuator						
Purpose	*This property set will be used by an IfcControlElement object to define the characteristics of a hydraulic actuator*					
ManualOverride	*Identifies whether hand-operated*	*BOOL*	*FALSE*	*TRUE*	*FALSE*	*n/a*

	operation is provided as an override					
InputPressure	Maximum design pressure for the actuator	IfcPressureMeasure	see type	see type	0.0	n/a
InputFlowrate	Maximum hydraulic flowrate requirement	IfcVolumetricFlowrateMeasure	see type	see type	0.0	n/a
Pset_HandOperatedActuator						
Purpose	This property set will be used by an IfcControlElement object to define the characteristics of a hand operated actuator					
ManualOverride	Identifies whether hand-operated operation is provided as an override	BOOL	FALSE	TRUE	FALSE	n/a
Design Criteria						
Pset_DuctDesignCriteria						
Purpose	This property set will typically be used in conjunction with Pset_Fluid and Pset_Insulation.					
DesignName	A name for the design values	IfcString	see type	see type	empty string	n/a
SizingMethod	Enumeration that identifies the methodology to be used to size system components	Enum [ConstantFriction, ConstantPressure, StaticRegain]	n/a	n/a	ConstantFriction	n/a
PressureClass	Nominal pressure rating of the system components	IfcPressureMeasure	see type	see type	0.0	n/a
LeakageClass	Nominal leakage rating for the system components	IfcPressureMeasure	see type	see type	0.0	n/a
FrictionLoss	The pressure loss due to friction per unit length	IfcPressureMeasure/IfcLengthMeasure	see type	see type	0.0	n/a
LiningType	The insulating lining type to be used	Ref[Pset_Insulation]	n/a	n/a	NIL	n/a
InsulationType	The insulation type to be used	Ref[Pset_Insulation]	n/a	n/a	NIL	n/a
ScrapFactor	Sheet metal scrap factor	REAL	0.0	0.0	0.0	n/a
DuctSealant	Type of sealant used on the duct and fittings	IfcString	see type	see type	empty string	n/a
MaximumVelocity	The maximum design velocity of the air in the duct or fitting	IfcLinearVelocityMeasure	see type	see type	0.0	n/a
AspectRatio	The default aspect ratio	REAL	0.0	0.0	0.0	n/a
MinimumHeight	The minimum duct height for rectangular, oval or round duct	IfcLengthMeasure	see type	see type	0.0	n/a
MinimumWidth	The minimum duct width for oval or rectangular duct	IfcLengthMeasure	see type	see type	0.0	n/a
Pset_DuctSystemDesignCriteria						
Purpose	This property set will typically be used in conjunction with Pset_Fluid and Pset_Insulation					
SystemType	Enumeration that identifies the type of system	Enum [VariableAirVolume, ConstantVolume, DoubleDuct]	n/a	n/a	VariableAirVolume	n/a
SystemDescription	System description	IfcString	see type	see type	empty string	n/a
SystemLocation	Physical description of the part of the building the system serves	IfcString	see type	see type	empty string	n/a
Pset_PipeDesignCriteria						
Purpose	This property set will typically be used in conjunction with Pset_Fluid and Pset_Insulation.					
DesignName	A name for the design values	IfcString	see type	see type	empty string	n/a
SizingMethod	Enumeration that identifies the sizing method to be used if different from the system design criteria	Enum [ConstantFriction, ConstantPressure]	n/a	n/a	ConstantFriction	n/a
PressureClass	Nominal pressure rating of the piping system components (i.e., 125, 250, etc.)	IfcPressureMeasure	see type	see type	0.0	n/a
MaximumVelocity	The maximum allowable fluid velocity	IfcLinearVelocityMeasure	see type	see type	0.0	n/a
InsulationType	The insulation type to be used	Ref[Pset_Insulation]	n/a	n/a	NIL	n/a
Pset_PipeSystemDesignCriteria						
Purpose	This property set will typically be used in conjunction with Pset_Fluid and Pset_Insulation					
SystemType	Enumeration that identifies the type of system	Enum [DomesticHotWater, ChilledWater, CondenserWater, HeatingHotWater, Steam]	n/a	n/a	ChilledWater	n/a
SystemDescription	System description	IfcString	see type	see type	empty string	n/a
SystemLocation	Physical description of the part of the building the system serves	IfcString	see type	see type	empty string	n/a
FluidSourcePressure	Pressure in main for domestic water, sprinklers, system pressure for hydronic systems, etc.	IfcPressureMeasure	see type	see type	0.0	n/a

FluidLiftHeight	*Lift that may be required on open systems with dense fluids*	*IfcPressureMeasure*	*see type*	*see type*	*0.0*	*n/a*
Physical Connection Sizes						
Pset_RectangularDuctConnection						
Purpose	*This property set provides size information about a rectangular duct connection*					
Width	*Width of rectangular duct*	*IfcLengthMeasure*	*see type*	*see type*	*0.0*	*n/a*
Height	*Height of rectangular duct*	*IfcLengthMeasure*	*see type*	*see type*	*0.0*	*n/a*
ConnectionType	*Enumeration that identifies the type of connection*	*Enum [DriveSlip, S-Slip, Flanged, SlipOn, StandingSeam, Angles, Other]*	*n/a*	*n/a*	*Flanged*	*n/a*
Pset_RoundDuctConnection						
Purpose	*This property set provides size information about a round duct connection*					
Diameter	*Diameter of round duct*	*IfcLengthMeasure*	*see type*	*see type*	*0.0*	*n/a*
ConnectionType	*Enumeration that identifies the type of connection*	*Enum [BeadedSleeve, Drawband, OutsideSleve, Flanged, Crimp, Swedge, Other]*	*n/a*	*n/a*	*Flanged*	*n/a*
Pset_OvalDuctConnection						
Purpose	*This property set provides size information about an oval duct connection*					
Width	*Width of oval duct*	*IfcLengthMeasure*	*see type*	*see type*	*0.0*	*n/a*
Height	*Height of oval duct*	*IfcLengthMeasure*	*see type*	*see type*	*0.0*	*n/a*
ConnectionType	*Enumeration that identifies the type of connection*	*Enum [BeadedSleeve, Drawband, OutsideSleve, Flanged, Crimp, Swedge, Other]*	*n/a*	*n/a*	*Flanged*	*n/a*
Pset_PipeConnection						
Purpose	*This property set provides size information about a pipe connection*					
NominalDiameter	*Nominal diameter of pipe*	*IfcLengthMeasure*	*see type*	*see type*	*0.0*	*n/a*
ConnectionType	*Enumeration that identifies the type of connection*	*Enum [Flanged, Screwed, Welded, BellAndSpigot, Threaded, Other]*	*n/a*	*n/a*	*Flanged*	*n/a*
CoordinationRequirement						
Pset_CoordinationRequirement						
Purpose	*This property set provides a placeholder for interoperable coordination requirements between different disciplines*					
OriginatingActor	*The actor which originates the coordination requirement*	*Ref [IfcActor]*	*see type*	*see type*	*NIL*	*n/a*
AffectedActor	*The actor which must act upon the coordination requirement*	*Ref [IfcActor]*	*see type*	*see type*	*NIL*	*n/a*
Requirement	*The coordination requirement*	*IfcString*	*see type*	*see type*	*empty string*	*n/a*

6.4. [BS-3] Pathway Design and Coordination

6.4.1. Object Types

None defined in this project

6.4.2. Type Definitions

None defined in this project

6.4.3. Property Sets

None defined in this project

6.5. [BS-4] HVAC Loads Calculation

6.5.1. Object Types

None defined in this project

6.5.2. Type Definitions

None defined in this project

6.5.3. Property Sets

None defined in this project

6.6. [CS-1] Code Checking - Energy Codes

6.6.1. Object Types

Class name			**OPT**ional, **INV**erse flags, **SELF** - redeclared relationship				
	Interface			**Data type/Related type**			
		Attribute/Relationship		**or Superclasses**	**Min**	**Max**	**Default**
IfcBuildingEnvelope				*IfcBuildingObject*			
	I_BuildingEnvelope						
		ThermalElements	OPT	Set[0:N] Ref[IfcLayeredElement]	see type	see type	NIL
		OccupancyType	OPT	IfcEnvelopeOccupancyTypeEnum	n/a	n/a	NIL
		InternalLoadDensity	OPT	IfcReal	0	10	0
		ThermalLoad	OPT	IfcReal	0	1.00E+16	0
IfcIntent				*IfcControlObject*			
	I_Intent						
		Source	IfcOwnerId	n/a	n/a	n/a	
		Description	IfcString	n/a	n/a	Unknown	
IfcLightingFixture				IfcFixture			
	I_LightingFixture						
		Category	OPT	Ref[IfcLightingFixtureTypeLibraryEntry]	n/a	n/a	n/a
		LampType	OPT	IfcLampTypeEnum	n/a	n/a	n/a
		LampDescription	OPT	IfcString	n/a	n/a	n/a
		WattagePerLamp	OPT	IfcInteger	0	2000	0
		NumberOfLampsPerFixture	OPT	IfcReal	0	4.00E+01	0
		FixtureIdentification	OPT	IfcString	n/a	n/a	n/a
		FixtureWattage	OPT	IfcInteger	0	3000	0
		BallastType	OPT	IfcBallastTypeEnum	n/a	n/a	n/a

		NumberOfFixtures	OPT	IfcInteger	0	1000	0
IfcLightingFixtureType							
	I_LightingFixtureType						
		Description	OPT	IfcString	n/a	n/a	n/a
		LampType	OPT	IfcLampTypeEnum	n/a	n/a	n/a
		LampDescription	OPT	IfcString	n/a	n/a	n/a
		WattagePerLamp	OPT	IfcInteger	0	2000	0
		BallastType	OPT	IfcBallastTypeEnum	n/a	n/a	n/a
IfcPropertyConstraint				IfcControlObject			
	I_PropertyConstraint						
		Source	OPT	IfcOwnerId	n/a	n/a	n/a
		ReferenceObject	OPT	IfcProjectObject/ IfcAttributeObject	n/a	n/a	NIL
		Relation	OPT	IfcNumericRelation	see type	see type	NIL
		ConstraintType	OPT	IfcConstraintLevel	see type	see type	NIL
		NoticeText	OPT	IfcString	n/a	n/a	NIL

6.6.2. Type Definitions

Type
IfcEnvelopeOccupancyTypeEnum
IfcLightingOccupancyType
IfcLightingScopeEnum

6.6.3. Property Sets

PreDefined PropertySets in CS1 Code Checking - Energy Codes							
	PropertySet (Pset) Name						
		Attribute or Relation name	**Definition**	**Data Type or Related Object**	**Min**	**Max**	**Default**
Shared PropertySets defining Type							
	(** No SharedPropertySets defined in this schema **)						
Type driven PropertySets that vary for each occurrence							
	Pset_SystemLighting						
		LightingScope		IfcLightingScopeEnum	See type	See type	Zone
		SystemElements	Contains references to all instances of IfcFixture that are part of the lighting system	Ref[IfcFixture]	see type	see type	NIL
		OccupancyType	Lighting occupancy type according to the Standard	IfcLightingOccupancyType	n/a	n/a	NIL

		LightingPowerDensity	Lighting power density specified by the Code (based on Occupancy type)	IfcReal	0	10	0
		LighingPower	Total lighting power for the proposed design	IfcReal	0	1.00E+16	0

6.7. [CS-2] Code Checking Extensions

6.7.1. Object Types

	Class Name	Interface name	Attribute / Relation name	Definition	Data Type or Related Object Type	Min	Max	Default	Units
					{{ "Ref" = relationship }}				
	IFC R2.0 CS-2 Project								
	--> Classes/Interfaces/Attributes/Relationships/Defined Types								
1	**IfcRamp**			*This class provide an inclined way connecting one level to another.*					
			InheritsFrom ---------->	*IfcAssembledElement*					
		I_Ramp							
			hasElements	*list of IfcRampElement which can be a flight or landing*	*List [1:N] IfcRampElement*	*n/a*	*n/a*	*NIL*	*n/a*
2	**IfcRampElements**								
			InheritsFrom ---------->	*IfcBuildingElement*					
		I_RampElements							
			RampElementsType	*Predefined generic types of ramp element which can be a landing or a flight*	*Ref[IfcTypeDefinition]*	*n/a*	*n/a*	*NIL*	*n/a*
			hasFootPath	*center line of the path of IfcRampElement*	*IfcPolyCurve3D*	*n/a*	*n/a*	*NIL*	*n/a*
			hasElementOutline	*closed 3D curve profile to describe the outline of the IfcRampElement*	*IfcCurve3D*	*n/a*	*n/a*	*NIL*	*n/a*
			hasSideElements	*side element is building object that attached to ramp element e.g. wall, column, balustrade*	*List [2:N] IfcSideElement*	*n/a*	*n/a*	*NIL*	*n/a*
			hasEffectiveWidth	*Minimum clear width of the ramp element*	*IfcReal*	*n/a*	*n/a*	*NIL*	*n/a*
			hasFloorMaterial	*Predefined generic types of approved ramp material which can be clay tiles, carpet, rubber sheet*	*Ref[IfcTypeDefinition]*	*n/a*	*n/a*	*NIL*	*n/a*
3	**IfcLanding**								
			InheritsFrom ---------->	*IfcBuildingElement*					
		I_Landing							
			hasEffectiveLength	*Minimum clear length for packing a stationary wheelchair on the landing*	*IfcReal*	*n/a*	*n/a*	*NIL*	*n/a*

	hasFloorLevel	*Floor level of A landing*	*IfcReal*	*n/a*	*n/a*	*NIL*	*n/a*
4	**IfcFlight**						
	InheritsFrom ---------->	*IfcBuildingElement*					
	I_Flight						
	hasVerticalRise	*change in floor level*	*IfcReal*	*n/a*	*n/a*	*NIL*	*n/a*
	hasHorizontalRun	*length of the run*	*IfcReal*	*n/a*	*n/a*	*NIL*	*n/a*
5	**IfcSideElement**						
	InheritsFrom ---------->	*IfcBuildingElement*					
	I_SideElement						
	SideElementType	*Predefined generic types of side elements e.g. Enum (wall, column, balustrade*	*Ref[IfcTypeDefinition]*	*n/a*	*n/a*	*NIL*	*n/a*
	hasBaluster	*baluster can also be handrails, railing*	*IfcBalusterType*	*n/a*	*n/a*	*NIL*	*n/a*
6	**IfcBaluster**						
	InheritsFrom ---------->	*IfcBuildingElement*					
	I_Baluster						
	BalusterType	*Predefined generic types of baluster or handrail*	*Ref[IfcTypeDefinition]*	*n/a*	*n/a*	*NIL*	*n/a*
	hasProfile	*Polycurve that defines the Path of baluster*	*IfcPolyCurve3D*	*n/a*	*n/a*	*NIL*	*n/a*
	hasGrippingArea	*Gripping area of the baluster*	*IfcArea*	*n/a*	*n/a*	*NIL*	*n/a*
7	**IfcLift**						
	InheritsFrom ---------->	*IfcEquipment*					
	I_Lift						
	LiftType	*Predefined types of lift e.g. Enum (Disabled, Cargo, Fire etc*	*Ref[IfcTypeDefinition]*	*n/a*	*n/a*	*NIL*	*n/a*
	hasEffectiveWidth	*Minimum clear width for the maneuvering of wheelchair into the lift*	*IfcReal*	*n/a*	*n/a*	*NIL*	*n/a*
	hasEffectiveLength	*Minimum clear length for packing A stationary wheelchair in the lift*	*IfcReal*	*n/a*	*n/a*	*NIL*	*n/a*
	hasEffectiveTurningArea	*Minimum area for turning of A wheelchair in the life and at the doorway of the lift*	*IfcArea*	*n/a*	*n/a*	*NIL*	*n/a*
	ServingStore	*Stories being served by the lift*	*List [1:N] IfcBuildingStorey*	*n/a*	*n/a*	*NIL*	*n/a*
8	**IfcSymbol**						
	InheritsFrom ---------->	*IfcEquipment*					
	I_Symbol						
	SymbolType	*Predefined types of symbol e.g. Enum (Disabled, Fire etc)*	*Ref[IfcTypeDefinition]*	*n/a*	*n/a*	*NIL*	*n/a*
	Placement	*Position of symbol*	*IfcPoint*	*n/a*	*n/a*	*NIL*	*n/a*
9	**IfcExitFacility**						
	InheritsFrom ---------->	*IfcSpace*					
	I_ExitFacility						
	ExitFacilityType	*Predefined types of symbol e.g. Enum (Disabled, Fire etc)*	*Ref[IfcTypeDefinition]*	*n/a*	*n/a*	*NIL*	*n/a*

	hasFireRating	*Number of hours*	*IfcReal*	*n/a*	*n/a*	*NIL*	*n/a*
	hasCapacity	*Maximum numbers of occupant passing through the facility at any one time*	*IfcInteger*	*n/a*	*n/a*	*NIL*	*n/a*
	hasEffectiveWidth	*Effective width of the facility*	*IfcReal*	*n/a*	*n/a*	*NIL*	*n/a*
10	**IfcExitApproach**						
	InheritsFrom ---------->	*IfcSpace*					
	I_ExitApproach						
	ExitApproachType	*Predefined generic types of exit approach e.g. Enum (e.g. exit passageway, exit corridor, exit lobby, exist staircase).*	*Ref[IfcTypeDefinition]*	*n/a*	*n/a*	*NIL*	*n/a*
	OtherRelatedUsage	*Predefined generic types of other usage for exit approach e.g. Enum (e.g. area of refuge, area of fire fighting).*	*List [1:n] of [IfcTypeDefinition]*	*n/a*	*n/a*	*NIL*	*n/a*
	HasFireResistanceProvision	*list of fire resistance provision*	*List [1:N] IfcFireResistanceProvision*	*n/a*	*n/a*	*NIL*	*n/a*
	HasOpening	*list of openings*	*List [1:N] IfcOpening*	*n/a*	*n/a*	*NIL*	*n/a*
11	**IfcFireResistanceProvision**						
	InheritsFrom ---------->	*IfcEquipment*					
	I_FireResistanceProvision						
	FireResistanceProvisionType	*- Predefined generic types of fire resistance provision e.g. Enum (e.g. smoke-free, sprinklered, naturally ventilated, mechanically ventilated equipment).*	*Ref[IfcTypeDefinition]*	*n/a*	*n/a*	*NIL*	*n/a*
12	**IfcDoor**						
	InheritsFrom ---------->	*IfcFillingElement*					
	I_Door						
	OpenType	*Predefined generic types to describe how door is opened (e.g. Sliding, Swing, one-way or two way)*	*Ref[IfcTypeDefinition]*	*n/a*	*n/a*	*NIL*	*n/a*
	SwingDirection	*Predefined generic types to describe the swing directions with respect to A pivot (pull and push Side of the swing to be indicated)*	*Ref[IfcTypeDefinition]*	*n/a*	*n/a*	*NIL*	*n/a*
	ClearanceSpace	*area of a square bounding the swing of a door*	*IfcArea*	*n/a*	*n/a*	*NIL*	*n/a*
13	**IfcBuilding**						
	InheritsFrom ---------->	*IfcBuildingObject*					
	I_Building						
	BuildingUsage	*- Predefined generic types to describe usage of building e.g. Enum (commercial, residential). This will in turn links to the occupancy load*	*Ref[IfcTypeDefinition]*	*n/a*	*n/a*	*NIL*	*n/a*
	OccupancyLoad	*Estimated maximum numbers of peoples likely to occupy the building at any one time*	*IfcInteger*	*n/a*	*n/a*	*NIL*	*n/a*
14	**IfcZone**						
	InheritsFrom ---------->	*IfcSpaceElement*					
	I_Zone						
	ZoneUsage	*Add to Predefined type of IfcZoneUsage to cover Fire compartment or zone with Fire-protected provision (including smoke, sprinkler) and disabled access provision*	*Ref[IfcTypeDefinition]*	*n/a*	*n/a*	*NIL*	*n/a*
	OccupancyLoad	*Estimated maximum numbers of peoples likely to occupy the zone at any one time*	*IfcInteger*	*n/a*	*n/a*	*NIL*	*n/a*

15	**IfcBuildingStorey**						
	InheritsFrom ---------->	*IfcBuildingObject*					
	I_BuildingStorey						
	OccupancyLoad	*Estimated maximum numbers of peoples likely to occupy the storey at any one time*	*IfcInteger*	*n/a*	*n/a*	*NIL*	*n/a*
	hasFloorLevel	*Floor level of A landing*	*IfcReal*	*n/a*	*n/a*	*NIL*	*n/a*
	hasExitFacility	*list of exit facilities that found in the storey*	*List [1:N] IfcExitFacility*	*n/a*	*n/a*	*NIL*	*n/a*
16	**IfcSpace**						
	InheritsFrom ---------->	*IfcSpaceElement*					
	I_Space						
	Spacetype	*Add to Predefined type of IfcSpaceType to cover concept of internal and external with respect to the building envelop.*	*Ref[IfcTypeDefinition]*	*n/a*	*n/a*	*NIL*	*n/a*
	OccupancyLoad	*Estimate number of people likely to occupy a space at any one time.*	*IfcInteger*	*n/a*	*n/a*	*NIL*	*n/a*
	SpaceUsage	*Named types to describe usage of A space (e.g. classroom, staircase) This will in turn links to the occupancy load.*	*Ref[IfcTypeDefinition]*				
	hasExitFacility	*list of exit facilities that found in the storey*	*List [1:N] IfcExitFacility*	*n/a*	*n/a*	*NIL*	*n/a*

6.7.2. Type Definitions

IfcFireProtectionProvisionEnum	
	smokeFree
	sprinklered
	naturallyVentilated
	mechanicallyVentilated

6.7.3. Property Sets

PreDefined PropertySets in CS1 Code Checking - Energy Codes						
PropertySet (Pset) Name						
	Attribute or Relation name	**Definition**	**Data Type or Related Object**	**Min**	**Max**	**Default**
Pset_SystemLighting						
	LightingScope		IfcLightingScopeEnum	See type	See type	Zone
	SystemElements	Contains references to all instances of IfcFixture that are part of the lighting system	Ref[IfcFixture]	see type	see type	NIL
	OccupancyType	Lighting occupancy type according to the Standard	IfcLightingOccupancyType	n/a	n/a	NIL
	LightingPowerDensity	Lighting power density specified by the Code (based on Occupancy type)	IfcReal	0	10	0
	LighingPower	Total lighting power for the proposed design	IfcReal	0	1.00E+16	0

Pset_FireCompartment						
	ZoneUsage	*Add to Predefined type of IfcZoneUsage to cover Fire compartment or zone with Fire-protected provision (including smoke, sprinkler) and disabled access provision*	*Ref[IfcTypeDefinition]*	*n/a*	*n/a*	*NIL*
	OccupancyLoad	*Estimated maximum numbers of peoples likely to occupy the zone at any one time*	*IfcInteger*	*n/a*	*n/a*	*NIL*
Pset_Signage						
	SignType	*SignType*	*Predefined types of symbol e.g. Enum (Disabled, Exit, Fire Extinguisher, etc)*	*Ref[ifcRequiredSignage]*	*n/a*	*n/a*

6.8. [ES-1] Cost Estimating

6.8.1. Object Types

None defined in this project

6.8.2. Type Definitions

None defined in this project

6.8.3. Property Sets

None defined in this project

6.9. [FM-3] Property Management (Building Owner's viewpoint)

6.9.1. Object Types

Extensions to Core Model						
--> Classes/Interfaces/Attributes/Relationships/Defined Types						
IfcGrouping	*A number of object is selected from the IFC model either by IFC Objects type/attribute or by the user picks a number of objects from the model*					
InheritsFrom ---------->	*IfcControlObject*					
Description	*Description of Grouping object. Description can be used to identified purpose of the group.*	*IfcString*	*n/a*	*n/a*	*empty string*	*n/a*
Extensions to FM Model						
IFCMaintenanceObject						
InheritsFrom ---------->	*IfcControlObject*					
Description	*Description of maintenance operation.*					
Delivery date	*Time from which the guarantees start*	*IfcDate*	*n/a*	*n/a*	*n/a*	*n/a*

Guarantee terms (Pointer to)	*Guarantee terms. Descriptions of the right treatment of the product*	*?*	*n/a*	*n/a*	*empty string*	*n/a*
Guarantee ending date	*Date where the guarantee ends*	*IfcDate*	*n/a*	*n/a*	*n/a*	*n/a*
Maintenance period	*Period between each maintenance operation*	*IfcDate*	*n/a*	*n/a*	*n/a*	*n/a*
Last maintenance date	*Last time maintenance was made.*	*IfcDate*	*n/a*	*n/a*	*n/a*	*n/a*
Maintenance handling	*Actor who has made the last maintenance*	*IfcActor*	*see type*	*see type*	*0,0*	*see type*
Maintenance Instruction (Pointer to)	*Link to external information about maintenance instructions. Information may be on paper or in a electronic form.*	*?*	*n/a*	*n/a*	*n/a*	*n/a*
Maintenance history (Pointer to	*Link to external information about the maintenance history. Maintenance history describe when and who there did the maintenance, and maybe some additional information. Information may be on paper or in a electronic form.*	*?*	*n/a*	*n/a*	*n/a*	*n/a*
Inspection intervals	*Time between inspections.*	*IfcDate*	*n/a*	*n/a*	*n/a*	*n/a*
Condition report	*Description of the state of the component.*	*IfcString*				
Last inspection date	*Last time inspection was made.*	*IfcDate*	*n/a*	*n/a*	*empty set*	*n/a*
Inspection handling	*Actor who has made the last inspection*	*IfcActor*	*n/a*	*n/a*	*n/a*	*n/a*
Inspection history (Pointer to)	*Link to external information about the last inspection. Information may be on paper or in a electronic form.*	*?*	*n/a*	*n/a*	*n/a*	*n/a*
Priority	*Importance of maintenance.*	*IfcInteger*	*0*	*N*	*0*	*n/a*
Cost	*Estimated cost for maintenance*	*IfcCost*	*n/a*	*n/a*	*n/a*	*n/a*

6.9.2. Type Definitions

None defined in this project

6.9.3. Property Sets

None defined in this project

6.10. [FM-4] Occupancy Planning

6.10.1. Object Types

Class Name							
Interface name		{{ "Ref" = reference }}					
Attribute / Relation name	**Definition**	**Data Type or Related Object Type**	**Min**	**Max**	**Default**	**Units**	**Optional**
Extensions to IFC Core Model							
--> Classes/Interfaces/Attributes/Relationships/Defined Types							
IfcSpaceRequirement	represents the summary of requirements of a space						
InheritsFrom ---------->	IfcControlObject						
I_SpaceRequirement							
SecurityRequirements	description of security requriments of a space	Set [0:N] of IfcString	n/a	n/a	empty set	n/a	
PrivacyRequirements	description of privacy requriments of a space	Set [0:N] of IfcString	n/a	n/a	empty set	n/a	
SpecialRequirements	any other types of requirements of a space	Set [0:N] of IfcString	n/a	n/a	empty set	n/a	

IfcPlan		general class for all types of plans for AEC/FM projects						
	InheritsFrom ---------->	IfcControlObject						
I_Plan								
	PlanID	identifier of the plan	IfcString	n/a	n/a	empty string	n/a	
	PlanName	name of the plan	IfcString	n/a	n/a	empty string	n/a	
	PlanDescription	general description of the plan	Set [0:N] of IfcString	n/a	n/a	empty set	n/a	
	PlanCreator	the authors of the plan	Set [0:N] of Ref. to IfcActor	n/a	n/a	empty set	n/a	
	CreationDate	the date that the plan is created	IfcDate	see type	see type	see type	see type	
I_PlanProject								
	Project	the project that the plan is created for	Ref. to IfcProject	see type	see type	NIL	see type	
I_PlanApproval								
	Approval	the approval information about the plan, such as approving persons, approving status, etc.	Ref. to IfcApproval	see type	see type	NIL	see type	
IfcOccupancySchedule		represents a space occupancy schedule						
	InheritsFrom ---------->	IfcControlObject						
I_OccupancySchedule								
	OccupyingActions	represents all the occupancy actions such as move actions and work tasks for the occupancy plan	Set [0:N] of Ref. to IfcProcessObject	n/a	n/a	empty set	n/a	
	PredsAndSuccs	sequencial relationships between the occupancy actions for the schedule	Set [0:N] of Ref. to IfcRelSequence	n/a	n/a	empty set	n/a	
	ScheduleData	all the time related scheduling data about the schedule such as start time, finish time, etc.	Att_ScheduleData	see type	see type	see type	see type	
I_OccupancyScheduleResponsible								
	Responsible	the person who is responsible for this schedule	Ref. to IfcActor	see type	see type	NIL	see type	
IfcMoveAction		represent each of the move of people						
	InheritsFrom ---------->	IfcProcessObject						
I_MoveActionElement								
	OccupantsToMove	people who are moving out or in the spaces	Set [0:N] of Ref. To IfcActor	n/a	n/a	empty set	n/a	Yes
	FF&EToMove	the furniture, fixture and equipment that are moved out or in the spaces	Set [0:N] of Ref. To IfcElement	n/a	n/a	empty set	n/a	Yes
I_MoveAction								
	MoveFrom	the space from which people or FF&E are moving out of	Ref. To IfcSpace	n/a	n/a	NIL	n/a	Yes
	MoveTo	the space to which people or FF&E are moving into	Ref. To IfcSpace	n/a	n/a	NIL	n/a	No
	Schedule	time related information for this move action such as start time, etc.	Att_ScheduleData	see type	see type	see type	see type	Yes
	Constraints	the constraints that tie this move action	Set [0:N] of IfcMoveActionConstraint	n/a	n/a	empty set	n/a	Yes
	Responsible	the person who is responsible for this action	Ref. To IfcActor	n/a	n/a	NIL	n/a	Yes
IfcMoveActionConstraint		the ocnstraint for a move action						
	InheritsFrom ---------->	IfcControlObject						
I_MoveActionConstraint								
	ConstraintType	type of the constraint such as as soon as possible, as late as possible, must move out by, etc.	IfcString	n/a	n/a	empty string	n/a	No
	ConstraintDate	the date requriement for certain constrainttype such as must move out by 'date', etc.	IfcDate	see type	see type	see type	see type	Yes
IfcMovePlan		the plan for moving people and FF&E						
	InheritsFrom ---------->	IfcPlan						
I_MovePlan								
	OccupancySchedule	the schedule for the move plan	Ref. To	n/a	n/a	NIL	n/a	Yes

				IfcOccupancySchedule					
		ProgramGroupToBeMoved	represents the requirement summary of the proposed spaces for the occupancy plan	Ref. To IfcProgrammeGroup	n/a	n/a	NIL	n/a	Yes
		RequiredWork	the work requried to change the spaces to meet the requirements defined in the ProgramGroupToBeMoved, such as a space modification, remodeling,etc.	Set [0:N] of IfcWorktask	n/a	n/a	empty set	n/a	Yes
		ProjectCostEstimate	the cost estimate for the move plan	IfcCostSchedule	see type	see type	see type	see type	Yes
	I_MovePlanDocument								
		Documents	all the general documents required and generated for the plan	Set [0:N] of Ref. To IfcDocument	n/a	n/a	empty set	n/a	Yes
		WorkOrders	work orders generated in the plan	Set [0:N] of Ref. To IfcDocument	n/a	n/a	empty set	n/a	Yes
		PurchaseOrders	puchase orders generated in the plan	Set [0:N] of Ref. To IfcDocument	n/a	n/a	empty set	n/a	Yes
		ChangeOrders	change orders generated for the plan for change of work	Set [0:N] of Ref. To IfcDocument	n/a	n/a	empty set	n/a	Yes
IfcApproval			represents general approval processes for a plan, a design, a proposal, a change order, etc.						
	I_Approval								
		ApprovalRequestFrom	person who requests the approval	Ref. To IfcActor	n/a	n/a	NIL	n/a	No
		ApprovalRequestTo	person who is asked for approval	Ref. To IfcActor	n/a	n/a	NIL	n/a	No
		ApprovalPersonForDesign	person who approves the design	Ref. To IfcActor	n/a	n/a	NIL	n/a	Yes
		RequestingDate	the date issuing the request	IfcDate	see type	see type	see type	see type	No
		RequestedDate	the date requested that approval need to be determined	IfcDate	see type	see type	see type	see type	Yes
		ApprovalDate	date that the result of the approval process is produced	IfcDate	see type	see type	see type	see type	Yes
		IsApproved	the result of the approval process, can be 'yes' or 'no'	IfcBoolean	see type	see type	see type	see type	Yes
		ApproveConstraint	additional constraints on a 'yes' approve	IfcConstraint	see type	see type	see type	see type	Yes
		ApprovalFor	represents the design, plan, or work order, etc. that the approval is for	Ref. to IfcProjectObject	n/a	n/a	NIL	n/a	Yes
IfcBudget			represents a cost budget for a project, a plan, etc.						
		InheritsFrom ---------->	IfcControlObject						
	I_Budget								
		BudgetCode	the code number for the budget	IfcString	n/a	n/a	empty string	n/a	No
		BudgetSource	description of the source of the budget	IfcString	n/a	n/a	empty string	n/a	Yes
		BudgetDescription	general description of the budget	Set [0:N] of IfcString	n/a	n/a	empty set	n/a	Yes
		BudgetTotal	the cost tatal of the budget	IfcCost	see type	see type	see type	see type	Yes
		BudgetBalance	difference between the BudgetTotal and the sum of the SubitemCosts	IfcCost	see type	see type	see type	see type	Yes
		OriginalBudgetAvailableDate	the date that the budget becomes available	IfcDate	see type	see type	see type	see type	Yes
		BudgetAvailableDuration	the time longevity of the budget	IfcTimeDuration	0	see type	0	see type	Yes
		Subitems	description of the subitems	List of IfcString	n/a	n/a	empty list	n/a	Yes
		SubitemCosts	costs of the subitems	List of IfcCost	n/a	n/a	empty list	n/a	Yes
	I_BudgetHistory								
		OriginalBudget	this allows tracking of the current budget status to date compared to the last previous budget	Ref. to IfcBudget	n/a	n/a	NIL	n/a	Yes
		UpdateDate	the date that this budet is updated; this allows tracking the budget usage history	IfcDate	see type	see type	see type	see type	Yes

IfcInventory	represents a general inventory information; supertype of space inventory, furniture inventory and equipment inventory						
I_Inventory							
InventoryDescription	general description of the inventory	IfcString	n/a	n/a	empty sting	n/a	Yes
InventoryScope	this can reference to a building, storey, or space	Ref. to IfcProductObject	n/a	n/a	NIL	n/a	No
InventoryJurisdiction	the organizational unit of the inventory	Ref. to IfcActor	n/a	n/a	NIL	n/a	No
InventoryResponsible	persons who are responsible for the inventory	Set [0:N] of Ref. to IfcActor	n/a	n/a	empty set	n/a	Yes
LastUpdateDate	date of last update	IfcDate	see type	see type	see type	see type	No
IfcDocument	represents a general document of anytype						
InheritsFrom ---------->	IfcControlObject, IfcProductObject						
I_Document							
GenericDocumentType	used to differentiate between an electronic and a paper document	IfcTypeDefinition	see type	see type	see type	see type	No
DocumentType	used to differentiate between a drawing and specification, and etc.	IfcTypeDefinition	see type	see type	see type	see type	No
IfcInteraction	represents interactive relationships between two IFC objects						
InheritsFrom ---------->	IfcControlObject						
I_Interaction							
Description	general description of the interaction	IfcString	see type	see type	empty sting	see type	Yes
InteractionType	the type of interaction, e.g. between actors. (see TypeDef)	IfcTypeDefinition	see type	see type	see type	see type	No
FrequencyDaily	number of interactions daily	IfcInteger	0	n/a	0	n/a	No
AverageDuration	average time duration of each interaction	IfcTimeDuration	0	see type	0	see type	Yes
ImportanceRating	represents the level of importance of interaction	IfcString	n/a	n/a	empty string	n/a	Yes

Extensions to IFC FM Model

--> Classes/Interfaces/Attributes/Relationships/Defined Types

IfcSpaceInventory	inventory for spaces						
InheritsFrom ---------->	IfcInventory						
I_SpaceInventory							
HasSpaces	all the spaces stored in the inventory; this allows to include zones	Set [1:N] of Ref. to IfcSpaceElement	n/a	n/a	NIL	n/a	No
TotalSpaces	total number of spaces in the inventory	IfcInteger	0	see type	0	n/a	No
TotalNetArea	total net area of all the spaces; can be derived from each space	IfcArea	see type	see type	see type	see type	Yes
IfcFurnitureInventory	inventroy for furniture						
InheritsFrom ---------->	IfcInventory						
I_FurnitureInventory							
TotolValueOriginal	the original total value of all the furniture	IfcCost	see type	see type	see type	see type	Yes
TotalValue	the current total value of all the furniture	IfcCost	see type	see type	see type	see type	Yes
FurnitureInventory	contains set of IfcFurniture and IfcWorkstation	Set [1:N] of Ref. to IfcElement	n/a	n/a	NIL	n/a	No
IfcEquipmentInventory							
InheritsFrom ---------->	IfcInventory						
I_EquipmentInventory							
TotolValueOriginal	the original total cost of all the equipment	IfcCost	see type	see type	see type	see type	Yes
TotalValue	the current total cost of all the equipment in the inventory	IfcCost	see type	see type	see type	see type	Yes

EquipmentInventory	all the equipment in the inventory	Set [1:N] of Ref. to IfcEquipment	n/a	n/a	NIL	n/a	No
IfcWorkstationWorkload	represents the workload of a workstation						
InheritsFrom ---------->	IfcControlObject						
I_WorkstationWorkload							
AverageWorkhourWeekly	average workhours each week	IfcTimeDuration	0	n/a	0	see type	Yes
TotalPaperfilesToStore	used to determine file storage	IfcInteger	0	n/a	0	see type	Yes
AveragePaperfilesProducedDaily	average total number of papers of paper files produced daily in the workstation	IfcInteger	0	n/a	0	see type	Yes
TotalComputerfiles	used to determine computer equipment, in unit of MB	IfcReal	0.0	n/a	0.0	see type	Yes
IfcWorkstationCompanyPolicy							
InheritsFrom ---------->	IfcControlObject						
I_WorkstationCompanyPolicy							
EmployeeType	e.g. manager, programmer, secretary, etc.	IfcString	n/a	n/a	empty string	n/a	No
MaxWorkstationSize	the maximum area of the workstation designed for the type of employee	IfcArea	see type	see type	see type	see type	Yes
MinWorkstationSize	the minimum area of the workstation designed for the type of employee	IfcArea	see type	see type	see type	see type	Yes
FurnitureStyle	the style of furniture for the workstation designed	IfcString	see type	see type	empty string	see type	Yes
CostLimit	the maximum cost limit for the workstation	IfcCost	see type	see type	see type	see type	Yes
IfcWorkstation							
InheritsFrom ---------->	IfcAssembledElement						
I_Workstation							
WorkstationComponents	list of worksurfaces and storages, tables, chairs, etc., excluding the vertical panels	Set [1:N] of Ref. to IfcManufacturedElement	n/a	n/a	empty set	n/a	Yes
WorkstationEquipment	all the equipment needed for the workstation	Set [1:N] of Ref. to IfcOfficeEquipment	n/a	n/a	empty set	n/a	Yes
WorkstationPanels	all the vertical panels for the workstation	Set [0:N] of Ref. to IfcSystemFurniture	n/a	n/a	empty set	n/a	Yes
Group	workstation group that the workstation belongs to	Ref. to IfcWorkstationGroup	n/a	n/a	NIL	n/a	Yes
I_WorkstationProgram							
Workload	the workload of the workstation	Ref. to IfcWorkstationWorkload	n/a	n/a	NIL	n/a	Yes
AssignedTo	the persons who are assigned to the workstation	Set [0:N] of Ref. to IfcActor	n/a	n/a	empty set	n/a	Yes
CompanyPolicy	the compancy policy that ties the design of the workstation	Ref. to IfcWorkstationCompanyPolicy	n/a	n/a	NIL	n/a	Yes
WorkstationInsideZones	2D zones inside of the workstation, e.g. work task zone, chair clear zone, cirrculation zone	Set [0:N] of Att_WorkstationZone2D	n/a	n/a	empty set	n/a	Yes
WorkstationRequirement	the space requiement for the workstation designed	Ref. to IfcSpaceRequiement	n/a	n/a	NIL	n/a	Yes
IfcSystemFurniture	represents any component of systems furniture such as worksurface, storage, etc.						
InheritsFrom ---------->	IfcManufacturedObject						
I_SystemFurnituer							
FurnitureType	Panel, Worksurface, Storage	IfcTypeDefinition	see type	see type	see type	see type	No
Workstation	the workstation composed with the component	Set [0:N] of Ref. to IfcWorkstation	n/a	n/a	NIL	n/a	Yes
IfcWorkstationGroup	represents a workstation group						
InheritsFrom ---------->	IfcAssembledElement						

I_WorkstationGroup								
	Workstations	all the workstations contained in the workstation group	Set [1:N] of Ref. to IfcWorkstation	n/a	n/a	empty set	n/a	Yes
	InFloorBlock	the 2D floor block that the workstation group covers	Ref. to IfcFloorBlock	n/a	n/a	NIL	n/a	Yes
	OnStorey	the storey that the workstation group located; should be used when floor block is not defined	Ref. to IfcStorey	n/a	n/a	NIL	n/a	Yes
	InSpace	the space containing the workstation group	Ref. to IfcSpace	n/a	n/a	NIL	n/a	Yes
	Profile	the 2D profile that represents the workstation boundary	IfcPolyCurve2D	see type	see type	see type	see type	Yes
	TotalArea	total area of the workstation group	IfcAreaMeasure	see type	see type	see type	see type	Yes
	WorkstationGroups	a workstation group can contain other group	Set [0:N] of Ref. to IfcWorkstationGroup	n/a	n/a	empty set	n/a	Yes
I_WorkstationGroupProgram								
	FunctionName	the function name of the workstation group, such as programming, marketing, etc.	IfcString	n/a	n/a	empty set	n/a	No
	Jurisdiction	the organizational unit that the workstation group belongs to	Ref. to IfcActor	n/a	n/a	NIL	n/a	No
I_WorkstationGroupSharedElement								
	SharedFurniture	shared furniture is not part of any workstations in the workstation group, e.g. a table for supporting a shared printe	Set [0:N] of Ref. to IfcManufacturedElement	n/a	n/a	empty set	n/a	Yes
	SharedEquipment	shared equipment is not part of any workstations in the workstation group, e.g. a shared printer	Set [0:N] Ref. to IfcOfficeEquipment	n/a	n/a	empty set	n/a	Yes
IfcFloorBlock		represents a 2D floor block						
	InheritsFrom ---------->	IfcControlObject						
I_FloorBlock								
	WorkstationGroups	workstation groups that the floor block covers	Set [0:N] of Ref. to IfcWorkstationGroup	n/a	n/a	empty set	n/a	Yes
	WorkstantionsNotInGroups	workstations that don't belong to a workstation group in the block	Set [0:N] of Ref. to IfcWorkstation	n/a	n/a	empty set	n/a	Yes
	InSpace	the space that contains the floor block	Ref. to IfcSpace	n/a	n/a	NIL	n/a	Yes
	CoverSpaces	the spaces contained in the floor block	Set [0:N] of Ref. IfcSpace	n/a	n/a	empty set	n/a	Yes
	Storey	the storey that the floor block is located	Ref. to IfcStorey	n/a	n/a	NIL	n/a	No
	Profile	the 2D profile of the floor block representing the boundary shape of the floor block	IfcPolyCurve2D	see type	see type	see type	see type	Yes
	Area	the area of the floor block	IfcAreaMeasure	see type	see type	see type	see type	Yes
I_FloorBlockProgram								
	FunctionName	the function name of the floor block such as marketing, or programming, etc.	IfcString	see type	see type	empty sting	see type	No
	Jurisdiction	organizational unit of the floor block.	Ref. to IfcActor	n/a	n/a	NIL	n/a	No

Extensions to Existing R1.5.1 Objects

IfcSpaceProgramme		{{ all attributes described in version 1.0 spec + the following }}						
	RequestedLocation	this can reference a building, storey, or space	Ref. to IfcProductObject	n/a	n/a	NIL	n/a	Yes
	ServiceRequirements	services required from the space	Set [0:N] of IfcString	n/a	n/a	empty set	n/a	Yes
	RequiredFF&E	required FF&E	Set [0:N] of Ref. to IfcTypeDefinition	n/a	n/a	empty set	n/a	Yes
	TargetDate	the target date of the space	IfcDate	see type	see type	see type	see type	Yes

								BudgetLimit	the budget limit for the space	IfcCost	see type	see type	see type	see type	Yes
								SpaceRequirements	the requirements for the space	Ref. to IfcSpaceRequirement	n/a	n/a	NIL	n/a	Yes

6.10.2. Type Definitions

#	TypeDef Name	Class being Typed	Generic Type	Specific Type / domain defining type	Att_Set #		Description
					Att_Set #		Attribute Sets
	DocumentType						
		IfcDocument					
			WorkOrder				
						shared =	Att_DocumentType
				Core		occurrence =	Att_WorkOrder
			PurchaseOrder				
						shared =	Att_DocumentType
				Core		occurrence =	Att_PurchaseOrder
			ChangeOrder				
						shared =	Att_DocumentType
				Core		occurrence =	Att_ChangeOrder
			Drawing				
						shared =	Att_DocumentType
				Core		occurrence =	Att_Drawing
			Specification				
						shared =	Att_DocumentType
				Core		occurrence =	Att_Specification
	GenericDocumentType						
		IfcDocument					
			ElectronicDocument				
						shared =	<none>
				Core		occurrence =	Att_ElectronicDocument
			PaperDocument				
						shared =	<none>
				Core		occurrence =	Att_PaperDocument
	FurnitureType						
		IfcSystemFurniture					
			Panel				
						shared =	Att_SystemFurnitureType
				FM		occurrence =	Att_Panel
			Worksurface				
						shared =	Att_SystemFurnitureType
				FM		occurrence =	Att_Worksurface
			Storage				
						shared =	Att_SystemFurnitureType

	FM	occurrence =	Att_Storage
InteractionType			
IfcInteraction			
ActorInteraction			
		shared =	<none>
	Core	occurrence =	Att_ActorInteraction

6.10.3. Property Sets

Attribute Set Name / Attribute or Relation name	Definition	Data Type or Related Object	Min	Max	Default	Units
Shared Attribute sets defining Type						
Core Model						
DocumentType						
Att_DocumentType	common attributes for all different types of documents	IfcDocument				
Author	the author of the document	IfcString	n/a	n/a	empty stirng	n/a
Company	the owner company of the document	IfcString	n/a	n/a	empty stirng	n/a
Title	the title of the document	IfcString	n/a	n/a	empty stirng	n/a
RevisionCode	revision code of the revision	IfcString	n/a	n/a	empty stirng	n/a
RevisionNumber	revision number of the revision	IfcInteger	0	see type	0	see type
LastModifiedDate	the date that the file last modified	IfcDate	see type	see type	see type	see type
FirstCreatedDate	the date that the file first created	IfcDate	see type	see type	see type	see type
FM Domain Extension Model						
FurnitureType						
Att_SystemFurnitureType	common attributes for all systems furniture types	IfcSystemFurniture				
Workstations	i.e. the workstations that the system furniture is assembled with or placed in as required by the design	set [0:n] of Ref. to IfcWorkstation	n/a	n/a	empty set	n/a
GroupCode	e.g. panels, worksurfaces, storages, etc.	IfcString	n/a	n/a	empty string	n/a
Width	i.e. nominal width	IfcLength	see type	see type	see type	see type
Height	i.e. nominal length	IfcLength	see type	see type	see type	see type
Finishing	e.g. walnut, fabric	IfcString	n/a	n/a	empty string	n/a
Type driven attributes that vary for each occurrence						
Core Model						
DocumentType						
Att_WorkOrder	occurrence attribute sets for WorkOrder typedef	IfcDocument				
TransactionCode	trasaction code	IfcString	n/a	n/a	empty string	n/a
RequestID	ID code of the request	IfcString	n/a	n/a	empty string	n/a
Facility	the building element that the work needs to be done for	Set [1:N] of Ref. to IfcProductObject	n/a	n/a	NIL	n/a
DateOfRequest	date of requested	IfcDate	see type	see type	see type	see type
ShortJobDescription	short description of the job requested	IfcString	n/a	n/a	NIL	n/a
JobDescription	description of the job requested	Set [0:N] of IfcString	n/a	n/a	empty set	n/a
Justification	the organizational unit that requests the job	Set [0:N] of IfcString	n/a	n/a	empty set	n/a
IfNotAccomplished	comments the job is not accomplished	Set [0:N] of IfcString	n/a	n/a	empty set	n/a
WorkRequest	work task requested	Ref. to IfcWorkTask	n/a	n/a	NIL	n/a
EstimatedCost	estimated cost	IfcCostSchedule	see type	see type	see type	see type
ContractualType	the contractual type of the work	Enum (InHouse, SelfHelp, Contract)	InHouse	Contract	InHouse	n/a

Budget	the budget requested	Ref. to IfcBudget	n/a	n/a	NIL	n/a
RequestBy	the person who requests the budget	Ref. to IfcActor	n/a	n/a	NIL	n/a
RequestTo	the person who receives the request	Ref. to IfcActor	n/a	n/a	NIL	n/a
AdditionalContact	additional contact person regarding the request	Ref. to IfcActor	n/a	n/a	NIL	n/a
Approval	the approval process of the	Ref. to IfcApproval	n/a	n/a	NIL	n/a
Att_PurchaseOrder	occurrence attribute sets for PurchaseOrder typedef	IfcDocument				
PurchaseOrderNo	the identification ID of the purchase order	IfcString	n/a	n/a	empty string	n/a
CompanyTitle	the company that issuesthe purchase order	Ref. to IfcActor	n/a	n/a	NIL	n/a
SupplierName	the supplier company	Ref. to IfcActor	n/a	n/a	NIL	n/a
Date	the date when the purchase order is issued	IfcDate	see type	see type	see type	see type
Remark	the remark comment	Set [0:N] of IfcString	n/a	n/a	empty set	n/a
DateRequired	the required date	IfcString	n/a	n/a	NIL	n/a
DateScheduled	the date scheduled	IfcDate	see type	see type	see type	see type
DateActual	actual date of receiving the items	IfcDate	see type	see type	see type	see type
FOB	Free of Board; 'yes' or 'no'	IfcBoolean	see type	see type	see type	see type
ShipMethod	method of shiping	IfcString	n/a	n/a	empty string	n/a
TotalCost	total cost of the purchase	IfcCost	see type	see type	see type	see type
TotalItems	total number of items to purchase	IfcInteger	0	see type	0	n/a
PurchaseItems	the items to purchase	List [0:N] of Att_PurchaseOrderItem	n/a	n/a	empty list	n/a
Approval	approval process information of the purchase order	Ref. to IfcApproval	n/a	n/a	NIL	n/a
Att_ChangeOrder	occurrence attribute sets for ChangeOrder typedef	IfcDocument				
ChangeOrderNo	the identification of the change order	IfcString	n/a	n/a	empty string	n/a
Description	general description of the change order	Set [0:N] of IfcString	n/a	n/a	empty set	n/a
Date	the date that the change order created	IfcDate	see type	see type	see type	see type
IssuedBy	the person who issued the change order	Ref. to IfcActor	n/a	n/a	NIL	n/a
IssuedTo	the person who receives the change order	Ref. to IfcActor	n/a	n/a	NIL	n/a
Approval	approval process information of the change order	Ref. to IfcApproval	n/a	n/a	NIL	n/a
Att_Drawing	occurance AttributeSet for DocumentType Drawing	IfcDocument				
DrawingId	the identification id of the drawing	IfcString	n/a	n/a	empty string	n/a
Specifications	the specification documents related to the drawing	Set [0:N] of Ref. to IfcDocument	n/a	n/a	empty set	n/a
Scale	the scale used for the drawing. E.g. if '1:100', the value is '100'	IfcReal	0.0	see type	0.0	see type
Unit	the measuring unit used for the drawing	IfcUnit	see type	see type	see type	see type
RelatedDrawings	other drawings that related to this drawing	Set [0:N] of Ref. to IfcDocument	n/a	n/a	empty set	n/a
Att_Specification	occurance AttributeSet for DocumentType Specificatioin	IfcDocument				
SpecificationId	the identification of the specification	IfcString	n/a	n/a	empty string	n/a
GeneralDescription	description of the specification	IfcString	n/a	n/a	empty string	n/a
RelatedDrawings	drawings related to the specification	Set [0:N] of Ref. to IfcDocument	n/a	n/a	empty set	n/a
TotalWords	total number of words	IfcInteger	0	see type	0	see type
Att_ElectronicDocument	occurance AttributeSet for GenericDocumentType ElectronicDocument	IfcDocument				
FileName	name of the file	IfcString	n/a	n/a	empty string	n/a
FileExtension_name	the extension name of the file	IfcString	n/a	n/a	empty string	n/a
Software	the software that creates the file	IfcString	n/a	n/a	empty string	n/a
FileSize	(in unit of KB)	IfcReal	0.0	see type	0.0	see type
Directory	the directory of the file	IftString	n/a	n/a	empty string	n/a
BackupFile	the backup version of the file	Ref. to IfcDocument	n/a	n/a	NIL	n/a
PaperCopy	the paper print copy of the file	Ref. to IfcDocument	n/a	n/a	NIL	n/a

				LastSaveTime	the time that the file last saved	IfcTime	see type	see type	see type	see type
				LastSaveDate	the date that the file last saved	IfcDate	see type	see type	see type	see type
				Type	hiden, readonly, etc.	IfcString	n/a	n/a	empty string	n/a
			Att_PaperDocument		occurance AttributeSet for GenericDocumentType Paper Document	IfcDocument				
				Location	more appropriate if there is something like IfcRootObject	Ref. to IfcProductObject	n/a	n/a	NIL	n/a
				TotalPages	total number of pages	IfcInteger	0	see type	0	see type
				ElectronicCopy	the electronic copy of the paper copy	Ref. to IfcDocument	n/a	n/a	NIL	n/a
			Att_Panel		occurance AttributeSet for FurnitureType Panel	IfcSystemFurniture				
				Shape	the vertical boundary shape of the panel	IfcPolyCurve2D	see type	see type	see type	see type
				Opening	an opening	IfcPolyCurve2D	see type	see type	see type	see type
				PanelType	e.g. Acoustical, Horz_Seg, Monolithic, Glazed, Open, Ends, Door, Screen, etc.	IfcString	n/a	n/a	empty string	n/a
				Thickness	the thickness of the panel	IfcLength	see type	see type	see type	see type
			Att_Worksurface		occurance AttributeSet for FurnitureType Worksurface	IfcSystemFurniture				
				UsePurpose	e.g. writing/reading, computer, meeting, printer, reference files, etc.	IfcString	n/a	n/a	empty string	n/a
				SupportType	i.e. Freestanding or supported	IfcString	n/a	n/a	empty string	n/a
				HungingHeight	the hunging height of the worksurface	IfcLength	see type	see type	see type	see type
				Thickness	the thickness of the worksurface	IfcLength	see type	see type	see type	see type
				ShapeDescription	corner square, rectangle, etc.	IfcString	n/a	n/a	empty string	n/a
			Att_Storage		occurance AttributeSet for FurnitureType Storage	IfcSystemFurniture				
				IsOverhead	is overhead storage or not	IfcBoolean	see type	see type	see type	see type
				SupportType	i.e. Freestanding or supported	IfcString	n/a	n/a	empty string	n/a
				UsePurpose	e.g. shelf, stationary, office supplies, personal items, etc.	IfcString	n/a	n/a	empty string	n/a
				NumberOfDrawers	number of drawers	IfcInteger	0	see type	0	see type
				HungingHeight	hunging height if IsOverhead	IfcLength	see type	see type	see type	see type
				Depth	depth of the storage	IfcLength	see type	see type	see type	see type
Extension Attribute sets										
			Att_ScheduleData		Attribute set for any object that uses schedule data set					
				TotalDuration	time duration	IfcTimeDuration	see type	see type	see type	see type
				ScheduledStartDate	scheduled start date	IfcDate	see type	see type	see type	see type
				ScheduledFinishDate	scheduled finish date	IfcDate	see type	see type	see type	see type
				ActualStartDate	actural start date	IfcDate	see type	see type	see type	see type
				ActualFinishDate	actural finish date	IfcDate	see type	see type	see type	see type
				EarlyStartDate	early start date	IfcDate	see type	see type	see type	see type
				EarlyFinishDate	early finish date	IfcDate	see type	see type	see type	see type
				LateStartDate	late start date	IfcDate	see type	see type	see type	see type
				LateFinishDate	late finish date	IfcDate	see type	see type	see type	see type
				TotalFloat	total float	IfcTimeDuration	see type	see type	see type	see type
				DaysRemaining	number of days remaining	IfcTimeDuration	see type	see type	see type	see type
			Att_PurchaseOrderItem		attribute set for purchase order items used in Att_PurchaseOrder	Att_PurchaseOrder				
				ItemNumber	the number of the purchase item in the list	IfcInteger	0	see type	see type	see type

	Quantity	quantity of the item	IfcReal	0.0	see type	0.0	see type
	Code	code of the item	IfcString	n/a	n/a	empty string	n/a
	Unit	unit describing the number of items	IfcUnit	see type	see type	see type	see type
	UnitPrice	unit price	IfcCost	see type	see type	see type	see type
	TotalCost	the cost of the item	IfcCost	see type	see type	see type	see type
	InvoiceAmount	cost amount of the itme on invoice	IfcCost	see type	see type	see type	see type
	TotalBalance	cost balance of the item	IfcCost	see type	see type	see type	see type
	InPurchaseOrder	the purchase order that the item is refered	Ref. to IfcDocument	n/a	n/a	NIL	n/a
Att_WorkstationZone2D		attribute set to represent functional 2D zones in IfcWorkstation	IfcWorkstation				
	WorkstationZoneType	e.g. worktask, circulation, chair_clearance, etc.	IfcString	n/a	n/a	empty string	n/a
	Length	length of the workstation	IfcLength	see type	see type	see type	see type
	Width	width of the workstation	IfcLength	see type	see type	see type	see type
Attribute sets that extend R1.5.1							
Att_FurnitureType		shared attribute set for all type of furniture	IfcFurniture				
	ProductCode	manufacture product code for the furniture type	IfcString	n/a	n/a	empty string	n/a
	Width	nominal overall width	IfcLength	see type	see type	see type	see type
	Height	nominal overall height	IfcLength	see type	see type	see type	see type
	Depth	nominal overall depth	IfcLength	see type	see type	see type	see type
	Material	the main material the furniture of this type is made of, e.g. walnut, etc.	IfcString	n/a	n/a	empty string	n/a
	Finishing	e.g. walnut, fabric	IfcString	n/a	n/a	empty string	n/a
Att_SpaceType		shared attribute set for all type of space	IfcSpace				
	SpaceName	the name of the space	IfcString	n/a	n/a	empty string	n/a
	GeneralDescription	general description of the space type	IfcString	n/a	n/a	empty string	n/a
	SpaceCatalog	description of space catalog	IfcString	n/a	n/a	empty string	n/a

6.11. [SI-1] Photo Accurate Visualization

6.11.1. Object Types

	Class Name	Interface name	Attribute / Relation name	OPT, INV, DER	Data Type/Related Object Type {{ "Ref" = relationship }}	Min	Max	Default	Definition
1	**IfcLightSource**								Luminary properties for a light source object. Note: geometry for the light source will be defined on the physical object which contains this object (e.g. IfcLightFixture)
		I_LightSource: SpectralPowerDistribution, PhotometricOutputDistribution							
			SpectralPowerDistribution		LIST [1:?] OF IfcMeasureWithUnit	n/a	n/a	n/a	List of Power/Wavelength value pairs
			PhotometricOutputDistribution		LIST [1:?] OF IfcPhotometricOutputSpace	n/a	n/a	n/a	List of Intensity/VolumeMeasure
2	**IfcPhotometricOutputSpace**								Space a luminaire to which a given average light intensity is radiated.
		I_PhotometricOutputSpace: OutputIntensity, OutputSpace							
			OutputSpace		IfcSpace	n/a	n/a	n/a	Space around luminaire (not the geometry will be placed relative to the luminaire
			OutputIntensity		IfcPhotometricMeasure	n/a	n/a	n/a	Light intensity for this space
3	**IfcLightFixture**								Type of electrical fixture which radiates light
		I_LightFixture: LuminaryProperties							
			LuminaryProperties		IfcLuminaire	n/a	n/a	n/a	Luminary properties for this light fixture
4	**IfcObjectMaterialSurface**								Planar surface on an object with a given material finish
		I_ObjectMaterialSurface: ObjectSurface, MaterialFinish							
			ObjectSurface		IfcSurface	n/a	n/a	n/a	Surface on an object with a consistent material and finish (as specified by MaterialFinish).
			MaterialFinish		IfcMaterialFinish	n/a	n/a	n/a	Material finish for this object surface.
5	**IfcMaterialFinish**								Finish applied to the surface of a material
		I_MaterialFinish							
			ForMaterial	INV	IfcMaterial	n/a	n/a	required	Pointer to the material to which this finish is applied. Note: each Material/Finish combination will have unique attributes
			FinishName		IfcString	n/a	n/a	empty string	Name for this material finish
			Reflectivity		IfcPositiveRatioMeasure	0.00	1.00	0.00	Measure for the ration of light reflected (versus absorbed) by this surface. Value of 0.00 means the value has not been set.
			Color		IfcString	n/a	n/a	empty string	Color of this surface, using the xxx color standard
			Texture		IfcString	n/a	n/a	empty string	Surface bumpiness - using the xxx standard
			BidirectionalScatteringDistribution		??	??	??	??	includes spectral reflectance and transmittance, specularity and roughness
			Polarization		??	??	??	??	??

6.11.2. Type Definitions

None defined in this project

6.11.3. Property Sets

None defined in this project

6.12. [XM-2]

6.12.1. Object Types

	Class Name								
		Interface name		OPT, INV, DER	{{ "Ref" = relationship }}				
	2 3 4 5 6		Attribute / Relation name		Data Type/Related Object Type	Min	Max	Default	Definition
1	**IfcDocumentTypeRegistry**								Reference to a project document
		I_DocumentTypeRegistry : DocumentTypes							
			DocumentTypes		LIST [0:?] OF IfcDocumentTypeDef	n/a	n/a	empty list	Semantic definitio for attribute 1
2	**IfcDocumentTypeDef**								Reference to a project document
		I_DocumentTypeDef : DocumentTypeID, Description							
			DocumentTypeID		STRING	n/a	n/a	n/a	Unique ID for this document type
			FileExtension		STRING [3]				File extension used by computer OS
			Description		STRING	n/a	n/a	n/a	Semantic definition for attribute 1
3	**IfcDocumentReference**								Reference to a project document
		I_DocumentReference : xxx							
			DocumentType		INTEGER	n/a	n/a	0	Index into DocumentTypeRegistry - identifying the type of document referenced. Zero indicates no type has been specified
			DocumentName		STRING	n/a	n/a	empty string	File name or document name assigned by owner
			Location		STRING	n/a	n/a	empty string	URL, pathname or physical location of the document
			DocSectionReference	OPT	STRING	n/a	n/a	empty string	Optional reference to a section within the document.
			DocumentOwner		INTEGER	n/a	n/a	0	Index into ProjectTeamRegistry - identifying the team member who "owns" this document. Zero indicates no owner has been specified
			ObjectSelectionSet		IfcObjectSelectionSet	n/a	n/a	empty list	selection set of objects "presented" in this document
4	**IfcObjectSelectionSet**								Reference to a project document
		I_ObjectSelectionSet : xxx							
			Objects		LIST [0:?] OF IfcObjectID	n/a	n/a	n/a	Semantic definitio for attribute 1
			ReferencingDocuments	INV	SET [0:?] OF IfcDocumentReferenc e	n/a	n/a	n/a	Semantic definitio for attribute 2

6.12.2. Type Definitions

None defined in this project

6.12.3. Property Sets

None defined in this project